**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**
**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**
**імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

*Кафедра автоматизованих систем обробки інформації і управління*

«На правах рукопису»

УДК _____

**«До захисту допущено»**
**В.о. завідувача кафедри**
_____ *О.А.Павлов*
(підпис) (ініціали, прізвище)
“___”_____2019 р.

# Магістерська дисертація

зі спеціальності *121 «Інженерія програмного забезпечення»*

на тему: «*Алгоритмічне та програмне забезпечення при визначенні авторства тексту»*

**Виконав:** студент VI курсу, групи *ІП-82мп*

*Щербаков Дмитро Сергійович* _____
(прізвище, ім’я, по батькові) (підпис)

**Науковий керівник** *доц., доц., к.т.н. Фіногенов О.Д.* _____
(посада, науковий ступінь, , вчене звання, прізвище та ініціали) (підпис)

**Консультант** *доц., к.т.н., Ліщук К.І.* _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

**Рецензент** доц., доц., к.т.н., Філіппова М.В. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2019 року

**Національний технічний університет України**
**«Київський політехнічний інститут**
**імені Ігоря Сікорського»**

Факультет *інформатики та обчислювальної техніки*
(повна назва)

Кафедра *автоматизованих систем обробки інформації та управління*
(повна назва)

Рівень вищої освіти *другий (магістерський) за освітньо-професійною програмою*

Спеціальність *121 «Інженерія програмного забезпечення»*
(код і назва)

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ *О.А. Павлов*
(підпис) (ініціали, прізвище)
«___» _____ 20*1* р.

**ЗАВДАННЯ**
**на магістерську дисертацію студенту**
*Щербакову Дмитру Сергійовичу*
(прізвище, ім'я, по батькові)

1. Тема дисертації *Алгоритмічне та програмне забезпечення при визначенні авторства тексту*

науковий керівник дисертації *к.т.н., доц. Фіногенов О.Д.*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від "____" _____ 20___ р. № _____

2. Строк подання студентом дисертації "____" _____ 20___ р.

3. Об'єкт дослідження *Художні тексти*

4. Предмет дослідження *Аналіз художніх текстів з метою визначення авторства*

5. Перелік завдань, які потрібно розробити *Дослідження праць авторів галузі; аналіз існуючих систем пошуку автора; розробка алгоритму; розробка програмного додатку; аналіз точності роботи*

6. Перелік графічного матеріалу

*Графік частотного розподілу грам авторів української літератури*

*Аналіз впливу імен головних героїв на розподіл грам*

7. Орієнтовний перелік публікацій *Статистичний аналіз художніх текстів українських авторів, Особливості використання грам при аналізі тексту та його перекладу*

8. Консультанти розділів дисертації

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|---|---|---|---|
| | | завдання видав | завдання прийняв |
| | | | |
| | | | |
| | | | |

9. Дата видачі завдання “ *01* ” *вересня* 20 *19* р

**Календарний план**

| № з/п | Назва етапів виконання магістерської дисертації | Строк виконання етапів магістерської дисертації | Примітка |
|---|---|---|---|
| 1 | *Дослідження наукової літератури* | *16.072019* | |
| 2 | *Аналіз існуючих систем пошуку автора* | *05.08.2019* | |
| 3 | *Постановка та формалізація математичної моделі задачі* | *10.08.2019* | |
| 4 | *Розробка алгоритму пошуку автора* | *07.09.2019* | |
| 5 | *Розробка програмного забезпечення* | *27.10.2019* | |
| 7 | *Проведення тестування алгоритму* | *14.11.2019* | |
| 8 | *Оформлення документації* | *01.12.2019* | |
| 9 | *Подання роботи на попередній захист* | *05.12.2019* | |
| 10 | *Подання роботи на основний захист* | *16.12.2019* | |
| | | | |

Студент _____ (підпис) _____ (ініціали, прізвище)

Науковий керівник _____ (підпис) _____ (ініціали, прізвище)

# РЕФЕРАТ

Магістерська дисертація: 86 с., 15 рис., 26 табл., 2 додатки, 15 джерел.

**Актуальність.** Ідентифікація та перевірка авторства є унікальною і, водночас, дуже широко розповсюдженою задачею, з огляду на можливість застосування у різних сферах діяльності людини: для боротьби з плагіатом, для встановлення авторства анонімних текстів, для експертизи та встановлення особистості в криміналістиці та у багатьох інших задачах та напрямах.

Проблеми дослідження стилю автора та застосування статистичного аналізу в дослідженні авторства розглядалася в дослідженнях О.О. Архипової та В.М. Журавльова, Л.А.Борисова, Ю.Н.Орлова та К.П. Осміна.

Задача є також дуже складною через фундаментальну проблему формування набору ознак, за якими можна оцінити ймовірність належності тексту певному автору.

Задачу ускладнює також той факт, що до останнього часу для розроблених систем визначення авторства текстів необхідною умовою їх стійкої та якісної роботи була наявність великих об’ємів авторських текстів у навчальній вибірці. Ще однією вадою розроблених моделей є їх якісне обмеження на кількість авторів.

**Мета дослідження.** Метою дослідження є розробка алгоритму та реалізація програмного забезпечення аналізу автора художнього тексту за допомогою методів статистичного аналізу, також встановлення впливу імен героїв та слів з великої літери в цілому на частотні характеристики тексту.

**Завдання дослідження.** Для досягнення мети необхідно виконати наступні завдання:

- дослідити праці авторів в цій галузі;
- проаналізувати існуючі системи пошуку автора;
- розробити алгоритм аналізу авторства;
- розробити програмний додаток заснований на розробленому алгоритмі;

- проаналізувати точність роботи алгоритму;
- зробити висновки щодо доцільності використання алгоритму.

**Об’єкт дослідження** – художні тексти.

**Предмет дослідження** – аналіз художніх текстів з метою визначення авторства.

**Методи дослідження.** Використовувалися методи статистичного аналізу для побудови алгоритму, методи об’єктно-орієнтованого програмування для розробки програмного додатку.

**Наукова новизна.** Найбільш суттєвими науковими результатами магістерської дисертації є:

- розроблений алгоритм дає високу точність при визначенні авторства художніх текстів;
- було досліджено вплив імен героїв та слів з великої літери в цілому на частотні характеристики тексту.

**Практичне значення.** Отримане програмне забезпечення дозволяє проводити аналіз художніх текстів та з високою точністю визначати автора тексту.

**Зв'язок роботи з науковими програмами, планами і темами.** Робота виконувалась на кафедрі автоматизованих систем обробки інформації та управління Національного технічного університету України «Київський політехнічний інститут ім. Ігоря Сікорського» в рамках теми «Математичні моделі та технології в СППР». Державний реєстраційний номер 0117U000914.

**Апробація.** Основні положення роботи доповідались і обговорювались на XVIII міжнародної науково-практичної конференції «Математичне та програмне забезпечення інтелектуальних систем МПЗІС-2019» та 3 всеукраїнської науково-практичної конференції молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2019).

**Публікації.** Наукові положення дисертації опубліковані в тезах XVIII міжнародної науково-практичної конференції «Математичне та програмне забезпечення інтелектуальних систем МПЗІС-2019» та в тезах 3 всеукраїнської науково-практичної конференції молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2019).

**Ключові слова:** Статистичний аналіз текстів, Визначення авторства.

## ABSTRACT

Master's Thesis: 86 pages, 15 figures, 26 tables, 2 appendixes, 15 sources.

**Actuality.** Authentication and verification is a unique and, at the same time, a very widespread task, given the possibility of being used in various fields of human activity: to combat plagiarism, to establish the authorship of anonymous texts, to assess and establish a person in criminology and many other tasks, and directions.

Problems of the study of author's style and application of statistical analysis in the study of authorship were considered in the researches of OO Archipova and V.M. Zhuravleva, LA Borisova, Yu.N.Orlov and K.P. Osmina.

The problem is also very complex because of the fundamental problem of forming a set of features that can be used to evaluate the likelihood of a text belonging to a particular author.

The proble is compounded by the fact that until recently, for the developed systems for determining the authorship of texts, a prerequisite for their stable and quality work was the availability of large volumes of copyrighted text in the training sample. Another drawback of the developed models is their qualitative limitation on the number of authors.

**The aim of the study.** The purpose of the study is to develop an algorithm and implement software for analysis of the author of the text with the help of statistical analysis methods, as well as to establish the influence of the names of characters and words in capital letters as a whole on the frequency characteristics of the text.

**Objectives of the study.** To achieve this goal, you must complete the following tasks:

- - to investigate the works of authors in this field;
- - analyze the author's existing search systems;
- - to develop an algorithm for analysis of authorship;
- - to develop a software application based on the developed algorithm;

- - analyze the accuracy of the algorithm;
- - to make conclusions about the expediency of eliminating the algorithm.

**The object of study** - the artistic texts.

**The subject of the study** is the analysis of artistic texts in order to determine authorship.

**Research methods.** We used statistical analysis methods to build the algorithm, object-oriented programming methods to develop a software application.

**Scientific novelty.** The most significant scientific results of the master's thesis are:

- - the developed algorithm gives high accuracy in determining the authorship of artistic texts;
- - the influence of the names of characters and capital letters as a whole on the frequency characteristics of the text was investigated.

**Practical meaning.** The resulting software allows you to analyze artistic texts and to determine the author of the text with high accuracy.

**Relationship with scientific programs, plans and topics.** The work was performed at the Department of Automated Information Processing and Management Systems of the National Technical University of Ukraine «Kyiv Polytechnic Institute. Igor Sikorsky ”within the topic“ Mathematical Models and Technologies in DSS ”. State Registration Number 0117U000914.

**Approbation.** The main provisions of the work were reported and discussed at the 18th International Scientific Conference “Mathematical and Software Software of Intelligent Systems IPAS-2019” and 3 All-Ukrainian Scientific and Practical Conference of Young Scientists and Students “Information Systems and Management Technologies” (ISTU-2019).

**Publications.** The scientific provisions of the dissertation are published in the theses of the XVIII International Scientific-Practical Conference "Mathematical and Software Software of Intelligent Systems IPAS-2019" and in the Theses of 3 All-Ukrainian

Scientific-Practical Conference of Young Scientists and Students "Information Systems and Technologies of Management" (ISTU-2019).

**Keywords:** Statistical analysis of texts, Definition of authorship.

# ЗМІСТ

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

Грама – умовна одиниця тексту при статистичному аналізі, тип грами зазвичай позначається цифрою перед словом грама: монограма, біграма, триграма тощо.

Частотний розподіл – розподіл, що враховує не кількість входів об’єкту в послідовність, а відносну частоту тобто відношення кількості входу об’єкто щодо кількості входу сумарно усіх об’єктів в масиві.

Функція щільності – функція, що, згідно досліджень, відображає стиль автора. Вираховується з урахуванням частотного розподілу грам у тексті.

# ВСТУП

Ідентифікація та перевірка авторства є унікальною і, водночас, дуже широко розповсюдженою задачею, з огляду на можливість застосування у різних сферах діяльності людини: для боротьби з плагіатом, для встановлення авторства анонімних текстів, для експертизи та встановлення особистості в криміналістиці та у багатьох інших задачах та напрямах.

Проблеми дослідження стилю автора та застосування статистичного аналізу в дослідженні авторства розглядалася в дослідженнях О.О. Архипової та В.М. Журавльова, Л.А.Борисова, Ю.Н.Орлова та К.П. Осміна.

Задача є також дуже складною через фундаментальну проблему формування набору ознак, за якими можна оцінити ймовірність належності тексту певному автору, крім того, роботу ускладнює також той факт, що до останнього часу для розроблених систем визначення авторства текстів необхідною умовою їх стійкої та якісної роботи була наявність великих об’ємів авторських текстів у навчальній вибірці. Ще однією вадою розроблених моделей є їх якісне обмеження на кількість авторів.

На сьогоднішній день з поширенням текстів завдяки мережі Інтернет, проблема дослідження авторства постає як ніколи гостро. З’являється така проблема, як так звані «літературні негри» - коли за автора текст пишуть менш талановиті письменники.

Також більшість сучасних систем перевірки на плагіат працюють переважно з науковими та технічними текстами, приділяючи значно менше уваги художнім.

Тому для визначення справжнього автора тексту часто доводиться звертатися до експертів, які можуть ідентифікувати автора невідомого тексту або визначити належність твору іншого автора за допомогою характерних мовних особливостей і різних стилістичних прийомів.

Отже, розроблення алгоритму визначення авторства художніх текстів та його програмна реалізація може значно полегшити правцю мовознавців, а отже є актуальною задачею сьогодення.

# 1 МЕТОДИ ВИЗНАЧЕННЯ АВТОРСТВА ТЕКСТУ

## 1.1 Огляд предметної області

Ідентифікація авторства тексту широко використовується у різних напрямах, зокрема, для експертизи у криміналістиці, для боротьби з плагіатом, встановлення авторства анонімного тесту тощо.

Завдання ідентифікації дуже складне. По-перше, виникає проблема формування набору ознак, за якими визначається ймовірність належності певного тексту окремому автору. По-друге, до останнього часу для якісної і точної роботи нині розроблених моделей визначення авторства необхідні великі об’єми авторських текстів у навчальній виборці. По-третє, якісне обмеження розроблених моделей на кількість авторів.

## 1.2 Аналіз досліджень предметній області

О.О. Архипової та В.М. Журавльова в своїй праці [1] провели дослідження частотного розподілу букв в українській мові. Вони зазначили, що такі характеристики, як розподіл букв та їх сполучуваність між собою є стійкими характеристиками мови. Згідно дослідження [2] частотні характеристики різних мов різняться між собою. Слід зазначити також, що частоти залежать не лише від мови та довжини тексту, а й від його типу, так, наприклад, у технічних текстах літера «Ф» може збільшити частоту через часте входження до технічних слів.

Ще більші відхилення від нормального розподілу спостерігаються в художніх творах, особливо в віршах.

В дослідженнях Л.А.Борисова, Ю.Н.Орлова та К.П. Осмініна [3,4,5] було доведено можливість визначення автора на прикладі творів російської літератури.

Статистичне визначення автора літературного тексту знаходиться у сфері інтересів літературознавців і математиків. У своїх творах письменники

дотримуються певної манери письма і це дозволить використовувати статистичні методи, щоб визначити належні йому тексти.

Завдання-максимум полягає в тому, щоб, механічно розібравши текст на певні елементи (в даному випадку буквосполучення), вказати його розташування в багатовимірному «фазовому» просторі світової літератури: мова, епоха написання, літературний напрям (течія, приналежність до певного кухоль і т.п.), тип (проза чи поезія), формат (роман, повість, нарис, есе), жанр (детектив, фантастика, трилер і т.д.), і, нарешті, автор. Це означає, що в ідеалі повинні бути визначені проектори на кожне підпростір в тій текстової структурі, яка зіставляється окремого твору і дозволяє з достатньою точністю провести аналіз. Такий текстової структурою в нашому методі є щільність функції розподілу тексту по буквосполученням або n-ПФР, де n відповідає порядку n-грам. Хоча для дискретних розподілів термін «ПФР» як щільність розподілу не цілком вдалий, він здався авторам більш прийнятним, ніж, наприклад, «емпірична частотна характеристика» тексту (ЕЧХ).

Основне припущення полягає в тому, що текст, написаний автором з метою передачі смислової інформації, містить в собі «проекцію» індивідуальності мислення автора. Тексти, написані одним і тим же автором, повинні володіти близькими «проекціями», а різними авторами - помітно відрізнятися, так що, якщо вдасться підібрати відповідне правило порівняння «проекцій» і, відфільтрувавши то загальне, що притаманне всім взагалі «Людям які пишуть», можна формально відрізнити одного учасника від іншого.

### 1.3 Методи аналізу авторства

Для визначення справжнього автора тексту звертаються до експертів, які можуть визначити належність твору конкретному автору, використовуючи характерні мовні особливості, різні стилістичні прийоми. На теперішній час для атрибуції текстів використовуються напрями теорії розпізнавання образів,

математичної статистики і теорії ймовірностей, алгоритми нейронних мереж, кластерний аналіз тощо.

Атрибуція тексту – це дослідження тексту з метою встановлення авторства або отримання будь-яких відомостей про автора і умови створення текстового документа. Завдання атрибуції можна розділити на ідентифікаційні і діагностичні [6]:

- ідентифікаційні завдання вирішуються з припущення, що автор тексту відомий;
- діагностичні завдання вирішуються з припущення, що автор тексту невідомий. Вони дозволяють визначити особистісні характеристики автора (рідна мова, освітній рівень, походження, місце постійного проживання тощо). А також дають змогу зафіксувати свідоме спотворення письмової мови;
- існують різноманітні методи аналізу стилю. Їх можна поділити на дві великі групи – експертні і формальні;
- експертні методи пропонують дослідження тексту професійним експертом-лінгвістом;
- формальні методи – використовують математичну статистику, теорію ймовірностей, алгоритм кластерного аналізу і нейронних мереж.

Існує 5 рівнів дослідження авторства:

- пунктуаційний рівень – особливості використання автором знаків пунктуації;
- орфографічний рівень – характерні помилки в написанні слів;
- синтаксичний рівень – особливості побудови речень у тексті;
- лексико-фразеологічний рівень – словниковий запас автора;
- стилістичний рівень – жанр, загальна структура тексту.

Також існує так званий класичний метод аналізу авторства. Класичний метод — це сукупність операцій, яка дозволяє інтерпретувати текст в цілях дослідника. Кожен документ створюється зі своєю, особливою метою, яка часто не збігається з цілями дослідження. Інтерпретація тексту в ході будь-якого традиційного аналізу дозволяє проникнути в сутність документа і виявити все, що цікавить дослідника, глибинні замисли та мотиви комунікатору, очікуваний ефект від повідомлення, особливості того історичного моменту, в який створювався документ тощо.

Види класичного методу аналізу:

- журналістський аналіз тексту - виділяє дві процедури такого аналізу: виявлення основних характеристик тексту та оцінка цих характеристик за визначеними критеріями. За характеристику приймаються тема, ідея, образний орієнтир [7];
- біографічний метод. За допомогою цього методу вчені досліджують розвиток суспільства і людини на прикладі конкретних життєписів. Об'єктом у таких дослідженнях є не тільки біографи чи автобіографії, а всі матеріали, за якими можна дослідити життєвий шлях людини. Цей метод застосовується в історичних, психологічних, психіатричних та багатьох інших науках. За допомогою цього методу вивчають настанову, мотиви поведінки особистості, роль різних соціальних прошарків у функціонуванні того чи іншого суспільства. До речі, в останній час при вивченні біографічних матеріалів вчені використовують і метод контент-аналізу [8]. Найбільш відоме в галузі біографічного аналізу дослідження В. Томаса та Ф. Знанецького «Польський селянин в Європі та Америці», написане в 1918-1920-х роках;
- історичний метод. Він включає в себе перевірку на дійсність тексту, яка встановлюється як за матеріальними ознаками, такими як папір, почерк, так і за формою тоном, стилем, словником автора тощо; з'ясування мотивів складання

документа, особи автора; висвітлення історичних обставин під час створення документа;

- літературний метод. За допомогою цього методу вивчаються стиль, тон, словник автора, композиція твору. Таким чином, вимальовуються індивідуальні творчі риси письменників, за якими можна довести, що саме їм належить чи не належить авторство тих або інших праць;

- лінгвістичний метод. Тут дослідника цікавить не зміст документа, а те, чи є в тексті морфеми, фонеми, яка частота вживання та розподіл лінгвістичних одиниць. Цей метод найбільше зі згаданих подібний до формалізованого методу контент-аналізу. Крім згаданих, існують ще психологічний, юридичний, соціологічний методи класичного аналізу офіційних документів та багато інших.

### 1.4 Огляд існуючих методів визначення авторства

Наведемо приклади існуючого програмного забезпечення з визначення авторства тексту.

Система «Лінгвоаналізатор». Метод, застосовуваний в цій системі для визначення авторства тексту, заснований на формальної математичної моделі. Програма враховує наступні характеристики мови автора: число службових слів; використовувані морфеми; рівень складності спожитих граматичних конструкцій; словниковий запас.

Система «Атрібутор». Дана програма є онлайн лінгвістичним процесором для машинного порівняння текстів і їх класифікації за параметрами індивідуального авторського стилю. Твори підбиралися так, щоб тексти різних письменників мали якомога більше відмінностей, а тексти одного письменника мали максимальні подібності. На цей момент система навчена порівнювати тільки тексти романів. Для атрибуції досить приблизно шість друкованих сторінок.

Система «СМАЛТ». Система складається з двох основних блоків: функціонального (аналіз, база даних) і аналітичного (Реалізація методик

статистичного аналізу текстів). Проект ще не допрацьований до кінця і передбачає розробку інформаційної системи, яка застосовує статистичні методи аналізу. В основі повинна бути база літературних творів, що складається з публіцистики 60-70 рр. 19 століття. Обробка текстів в даній системі проводиться поетапно. Спершу проводиться автоматизоване розбиття вихідного тексту на: розділ, абзац, пропозицію, слово. Потім здійснюється автоматична обробки тексту, його морфологічний розбір і синтаксичний аналіз. Після чого користувачем виконуються операції з бази даних з аналізу текстів.

Система «авторознавець». Програма, заснована на фоносемантическом аналізі, становить психологічний портрет автора. Система містить набір DLL-бібліотек, які підключаються до текстового процесора Word for Windows і в головному меню з'являється новий пункт. Таким чином, дана програмна система дозволяє користувачеві працювати в звичній для нього середовищі.

Серед програмних продуктів для визначення авторства текстів можна виділити систему «Антиплагіат» (http://www. antiplagiat.ru). Цей інтернет-сервіс пропонує здійснити перевірку текстових документів на наявність запозичень із загальнодоступних джерел. Система дозволяє проводити атрибуцію текстів на різних мовах. Пошук збігів здійснюється методом порівняння послідовностей символів без урахування мовних особливостей і мовних взаємозв'язків. За рахунок цього досягається висока, в кілька секунд, швидкість пошуку збігів.

Для виконання поставленого завдання застосовуються методи з теорії ймовірностей і математичної статистики для атрибуції текстів. Пропонований метод заснований на обліку статистики вживання пар елементів будь-якої природи, що йдуть один за одним у тексті (букв, морфем, словоформ і т. п.), т. е. на формальної математичної моделі послідовності букв (і будь-яких інших елементів) тексту як реалізації ланцюга Маркова. За темами творів авторів, які достовірно ними створені, обчислювалася матриця перехідних частот вживання пар елементів

(букв, граматичних класів слів і т. п.). Вона служила оцінкою матриці ймовірності переходу з елемента в елемент. Для кожного автора будувалася матриця перехідних частот і оцінювалася ймовірність того, що саме він написав анонімний текст (Або фрагмент тексту). Автором анонімного тексту вважався той, для кого розрахована оцінка ймовірності більше.

### 1.5 Статистичний аналіз

Для аналізу даних можуть застосовуватися різні методи. Статистичні методи аналізу даних призначені для їх ущільнення, виявлення взаємозв'язків і структур.

Статистичні методи - методи аналізу статистичних даних. За своєю природою вони поділяються на кількісні і категоріальні [9].

Кількісні (метричні) дані є безперервними за своєю структурою. Ці дані або виміряні за допомогою інтервальної шкали (числова шкала, кількісно рівні проміжки якої відображають рівні проміжки між значеннями вимірюваних характеристик), або за допомогою шкали відносин (крім відстані визначений і порядок значень).

Категоріальні (Неметричні) дані - це якісні дані з обмеженим числом унікальних значень і категорій. Існує два види категоріальних даних: номінальні - використовується для нумерації об'єктів і порядкові - дані, для яких існує природний порядок категорій.

Статистичні методи діляться на одно- і багатовимірні. Одномірні методи використовуються тоді, коли всі елементи вибірки оцінюються єдиним вимірником або якщо цих вимірників кілька для кожного елемента, але кожна змінна аналізується при цьому окремо від усіх інших [9].

Одномірні статистичні методи - методи статистичного аналізу даних у випадках, якщо існує єдиний вимірювач для оцінки кожного елемента вибірки або якщо ці вимірників декілька, але кожна змінна аналізується окремо від усіх інших.

Одномірні методи можна класифікувати на основі того, які дані аналізуються: метричні або неметричні. Метричні дані вимірюються по інтервального або відносній шкалі. Неметричні дані оцінюються за номінальною або порядкової шкалою. Потім ці методи ділять на класи на основі того, скільки вибірок - одна, дві або більше - аналізується в ході дослідження. Варто відзначити, що число вибірок визначається тим, як ведеться робота з даними для конкретного аналізу, а не тим, яким способом збиралися дані.

Варіація - це розходження у значеннях якої-небудь ознаки в різних одиниць даної сукупності в один і той же період або момент часу. Наприклад, працівники фірми розрізняються за доходами, витратами часу на роботу, росту, ваги, улюбленому заняттю у вільний час і т.д. Вона виникає в результаті того, що індивідуальні значення ознаки складаються під сукупним впливом різноманітних факторів (умов), які по-різному поєднуються в кожному окремому випадку. Таким чином, величина кожного варіанту об'єктивна [9].

Варіаційний ряд - це впорядкована розподіл одиниць сукупності найчастіше по зростаючим (рідше по убутним) значеннями ознаки і підрахунок числа одиниць з тим чи іншим значенням ознаки. Існують такі форми варіаційного ряду: ранжируваних ряд - являє собою перелік окремих одиниць сукупності в порядку зростання (або зменшення) досліджуваного ознаки; дискретний варіаційний ряд - таблиця, яка складається з конкретних значень варьирующего ознаки x та кількості одиниць сукупності з даним значенням f -ознака частот; інтервальний ряд - значення безперервного ознаки задаються інтервалами, які характеризуються інтервальної частотою т [9].

У процесі аналізу даних у дослідника регулярно виникає питання: чи достатньо значимі результати дослідження? Іншими словами, чи може результат пояснюватися тим, що у вибірку потрапили респонденти, які не представляють

генеральну сукупність в цілому? Для відповіді на це питання використовують статистичні гіпотези.

Гіпотези - це припущення або теорії, які дослідник висуває щодо деяких характеристик генеральної сукупності, що підлягає обстеженню. Користуючись статистичними прийомами, дослідник намагається встановити, чи існує емпіричне доказ, що підтверджує висунуті гіпотези. Перевірка статистичних гіпотез дозволяє розрахувати ймовірність настання якої-небудь події. Але в умовах відсутності повної всебічної інформації (що природно у випадках використання даних вибірки) завжди є певна ймовірність і помилкового висновку.

Висування гіпотези (нульовий або альтернативної). Нульова гіпотеза (Н 0), звана також гіпотезою status quo, являє собою твердження, в якому дослідник констатує факт відсутності будь-яких відмінностей яких впливів у вихідних даних. Вона призначена для визначення узгодженості вихідних даних з висунутим припущенням. Досліднику необхідно сформулювати нульову гіпотезу так, щоб відмова від неї приводив до бажаного висновку.

Альтернативна гіпотеза (Н а) призначена для визначення узгодженості даних з нульовою гіпотезою і спростовує її. У нашому прикладі проти нульової гіпотези можна висунути альтернативну гіпотезу виду Н а: Р> 0,20.

### 1.6 Постановка мети та задачі дослідження

Метою дослідження є розробка алгоритму та реалізація програмного забезпечення аналізу автора художнього тексту за допомогою методів статистичного аналізу, також встановлення впливу імен героїв та слів з великої літери в цілому на частотні характеристики тексту.

Об’єкт дослідження – статистичний аналіз тексту.

Предмет дослідження – аналіз художніх текстів з метою визначення авторства.

Задачі дослідження:

- дослідити праці авторів в цій галузі;

- проаналізувати існуючі системи пошуку автора;
- розробити алгоритм аналізу авторства;
- розробити програмний додаток заснований на розробленому алгоритмі;
- проаналізувати точність роботи алгоритму;
- зробити висновки щодо доцільності використання алгоритму.

### 1.7 Висновки

Ідентифікація та перевірка авторства є унікальною і, водночас, дуже широко розповсюдженою задачею, з огляду на можливість застосування у різних сферах діяльності людини: для боротьби з плагіатом, для встановлення авторства анонімних текстів, для експертизи та встановлення особистості в криміналістиці та у багатьох інших задачах та напрямах.

Було проведено аналіз наукових праць в предметній області, з них було винесено ідею аналізу тексту за допомогою статистичного аналізу тексту на основі розподілу грам, як одних з ключових особливостей стилю тексту.

Завдання встановлення авторства текстів (завдання атрибуції) зустрічається в різних областях і становить інтерес для філологів, літературознавців, юристів, криміналістів, істориків. В даний час для атрибуції текстів застосовуються: підходи з теорії розпізнавання образів, математичної статистики та теорії ймовірностей, алгоритми нейронних мереж і кластерного аналізу і багато інших.

Атрибуція тексту - дослідження тексту з метою встановлення авторства або отримання будь-яких відомостей про автора і умови створення текстового документа. Існує досить багато методів аналізу стилю. В цілому можна розділити їх на дві великі групи – експертні і формальні.

Існує досить багато методів аналізу стилю. В цілому можна розділити їх на дві великі групи – експертні і формальні:

- експертні методи передбачають дослідження тексту професійним лінгвістом-експертом;
- до формальних відносяться прийоми з теорії ймовірностей і математичної статистики, алгоритми кластерного аналізу і нейронних мереж.

Також існує так званий класичний метод аналізу авторства - — це сукупність операцій, яка дозволяє інтерпретувати текст в цілях дослідника.

Методи класичного аналізу тексту:

- журналістський, тобто аналіз основних характеристик аналізованого тексту;
- бібліографічний, що включає вивчення творів автора на протязі певного часу і виявлення певних ознак, характерних для автора, суспільства, або стилю автора на протязі часу;
- історичний, який включає аналіз тексту з перевіркою історичного контексту, перевірка чи міг текст бути написаним в досліджувані історичні рамки та за певних історичних обставин;
- літературний, вивчення основних ознак твору з точки зору літератури як науки: стиль, тон композиція;
- лінгвістичний, аналіз з урахуванням особливостей мовлення та побудови слів, словниковий запас, розподіл грам, аналіз пунктуаційних особливостей також стосується цього методу.

Метою дослідження є розробка алгоритму та реалізація програмного забезпечення аналізу автора художнього тексту за допомогою методів статистичного аналізу, також встановлення впливу імен героїв та слів з великої літери в цілому на частотні характеристики тексту. Об’єкт дослідження – статистичний аналіз тексту.

Об’єкт дослідження – художні тексти.

Предмет дослідження – аналіз художніх текстів з метою визначення авторства. Були поставлені наступні задачі для досягнення мети дослідження:

- дослідити праці авторів в цій галузі;
- проаналізувати існуючі системи пошуку автора;
- розробити алгоритм аналізу авторства;
- розробити програмний додаток заснований на розробленому алгоритмі;
- проаналізувати точність роботи алгоритму;
- зробити висновки щодо доцільності використання алгоритму.

# 2 МАТЕМАТИЧНИЙ РОЗДІЛ

## 2.1 Формалізація задачі аналізу

При проведені досліджень, а також при автоматичній обробці текстів часто виникає сукупність задач з ідентифікації автора тексту, жанру тексту, формату тексту (роман, повість, розповідь і т.д.) та інших. Одним з методів дослідження тексту є статистичний аналіз повторюваності елементів тексту (буквосполучень): грам, біграм, триграм. Кількісні характеристики тексту для кожного автора теоретично будуть відрізнятися в залежності від словникового запасу, жанру тексту, часу, коли текст був написаний та інших чинників. Ця умова не є достатньою. так як будь-який автор може скласти будь-який текст, але будемо вважати, що без накладання додаткових обмежень, авторський стиль буде давати змогу відрізнити автора від інших.

В таблиці 2.1 наведено частота символів російської та української мов, обрахована на множині текстів.

Таблиця 2.1 – Частота грамів російської та української мов

| № | Буква рос. мови | Частота | Буква укр. мови | Частота |
|---|---|---|---|---|
| 1 | А | 0.062 | А | 0.0807 |
| 2 | Б | 0.014 | Б | 0.0177 |
| 3 | В | 0.038 | В | 0.0535 |
| 4 | Г | 0.013 | Г, Ґ | 0.0155 |
| 5 | Д | 0.025 | Д | 0.0338 |
| 6 | Е, Ё | 0.072 | Е | 0.0495 |
| 7 | Ж | 0.007 | Ж | 0.0093 |
| 8 | З | 0.016 | З | 0.0232 |

Продовження таблиці 2.1

| | | | | |
|---|---|---|---|---|
| 9 | И | 0.062 | И | 0.0626 |
| 10 | Й | 0.010 | Й | 0.0138 |
| 11 | К | 0.028 | К | 0.0354 |
| 12 | Л | 0.035 | Л | 0.0369 |
| 13 | М | 0.026 | М | 0.0303 |
| 14 | Н | 0.053 | Н | 0.0681 |
| 15 | О | 0.090 | О | 0.0942 |
| 16 | П | 0.023 | П | 0.0290 |
| 17 | Р | 0.040 | Р | 0.0448 |
| 18 | С | 0.045 | С | 0.0424 |
| 19 | Т | 0.053 | Т | 0.0535 |
| 20 | У | 0.021 | У | 0.0336 |
| 21 | Ф | 0.002 | Ф | 0.0028 |
| 22 | Х | 0.009 | Х | 0.0119 |
| 23 | Ц | 0.004 | Ц | 0.0083 |
| 24 | Ч | 0.012 | Ч | 0.0141 |
| 25 | Ш | 0.006 | Ш | 0.0076 |
| 26 | Щ | 0.003 | Щ | 0.0056 |
| 27 | Ь, Ъ | 0.014 | І | 0.0575 |
| 28 | Ы | 0.016 | Ї | 0.0065 |
| 29 | Э | 0.003 | Ь | 0.0177 |
| 30 | Ю | 0.006 | Є | 0.0061 |
| 31 | Я | 0.018 | Ю | 0.0093 |
| 32 | Пробел | 0.175 | Я | 0.0248 |

Аналіз біграмів (буквосполучень 2-х символів) можна представити у вигляді температурної діаграми (рис. 2.1), де кольорами відображається частота використання, або у вигляді квадратної таблиці.

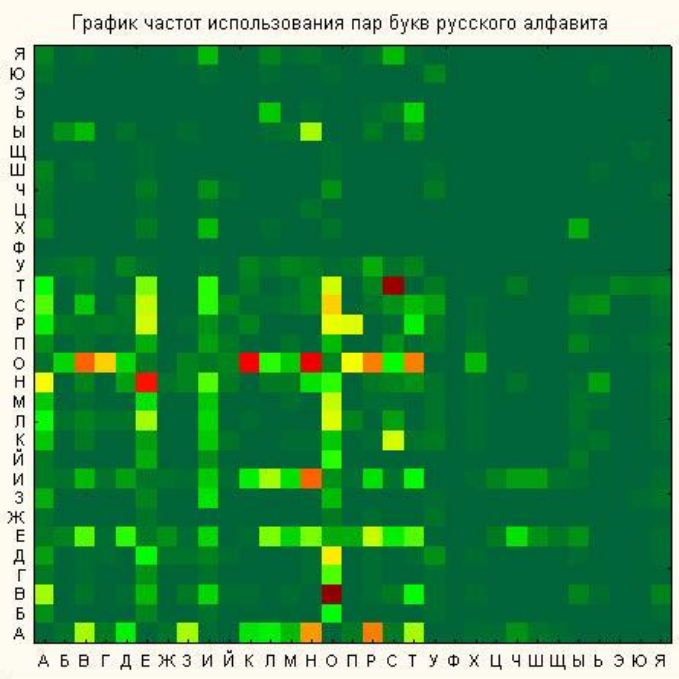



Рисунок 2.1 – Частоти використання пар букв російського алфавіту

Зазвичай, пробіли, розділові знаки, цифри, букви іншого алфавіту та інші символи ігноруються, якщо аналіз не направлений саме на ці особливості.

Формалізована задача розпізнавання автора тексту може бути представлена у наступний спосіб:

Нехай є бібліотека текстів, яка представлена у вигляді щільності функції розподілу або статистиці n-грамів (грамів, біграмів, триграмів і т.д.) для A відомих авторів. Нехай $K_a$ – кількість текстів a–го автора, $N_{i,a}$ – кількість символів в i-му тексті цього автора, де $i = 1,2,...,K_a$. Будемо вважати, що довжина кожного тексту достатня для проведення статистичного аналізу (тобто довжина тексту дає достатньо інформації про стиль автора в розрізі частоти зустрічі n-грамів).

Нехай $f_{i,a}(j)$ – функція щільності n-грамів відповідного тексту. де $j = 1,\ldots, a(n) = 1,\ldots,\Omega^n$ ($\Omega$ – розмір алфавіту або кількість символів, що досліджуються). Для кожного автора визначимо середньозважене значення щільності функції розподілу, нехтуючи відмінністю (одиницею) між грамами, біграмами і далі, внаслідок $N_a >> n$:

$$F_a(j) = \frac{1}{N_a}\sum_{i=1}^{K_a} f_{i,a}(j) N_{i,a}, \quad N_a = \sum_{i=1}^{K_a} N_{i,a}. \qquad (1)$$

Значення $F_a(j)$ – будемо вважати авторським еталоном.

Введемо «бібліотечну норму» $\rho_{ik}$ як відстань між щільністю функцій розподілу текстів $i$ та $k$ в нормі підсумованих функцій

$$\rho_{ik} = \|f_i - f_k\| = \sum_{j=1}^{\alpha(n)} |f_i(j) - f_k(j)|. \qquad (2)$$

Нехай в наявності є текст «0» невідомого автора, який необхідно ідентифікувати всередині даної бібліотеки. Автором тексту «0» вважається той з авторів «a», для якого норма $\rho_a^0 = \|f_0 - F_a\|$ різниці між щільністю функції розподілу $f_0(j)$ тексту «0» та середньою авторською щільністю функції розподілу $F_a(j)$ мінімальна:

$$\rho_a^0 = \|f_0 - F_a\|, \quad a^0 = \arg\min_a \rho_a^0. \qquad (6)$$

**2.2 Проблеми та метод їх усунення**

Першою проблемою при аналізі тексту подібними методами може бути наявність у тексті зайвих символів, або символів іншою мовою. Для вирішення

цього необхідно буде реалізувати модуль, що буде вилучати з тексту подібні символи, або модифікувати алгоритм таким чином, щоб він опрацьовував лише символи, що підлягають аналізу.

Другою проблемою постає необхідність збереження даних для їх подальшого аналізу. Необхідно буде не просто виводити дані, а й зберігати їх для подальшого аналізу та створення бази авторів. Вирішення цієї проблеми необхідно виконати на етапі розробки програмного додатку.

Третя проблема – необхідність можливості швидкого наочного аналізу даних для тих випадків, коли результат потрібен відразу, без подальшої обробки. Необхідно буде розробити метод відображення результатів аналізу тексту.

### 2.3 Побудова алгоритму

Таким чином, алгоритм пошуку автора складається з 4 кроків:

а) розбити текст на грами;

б) провести частотний аналіз тексту;

в) визначити відхилення для авторів у базі даних;

г) визначити автора з найменшим відхиленням, що і буде шуканим автором.

Перший крок необхідний, адже аналіз тексту проводиться програмно, і саме у такому вигляді він є найбільш сприйнятливий для методів машинної обробки даних.

Другий крок знаходить щільність функції розподілу тексту. Для цього рахуються входи грам, і після обробки усього тексту вираховуються частоти грами. Масив з входів усіх грам, що зустрічаються в тексті і буде шуканою функцією розподілу для тексту.

Необхідно також зазначити, що програмі необхідно ігнорувати символи, що не є грамами, тобто необхідний алгоритм фільтрації небажаних символів.

Третій крок забезпечить нормальну роботу четвертого, для цього нам необхідно отримати дані середніх розподілів по авторам, що є в базі, скласти для

них функції розподілу. Необхідно буде використовувати швидку базу даних на етапі проектування програмного додатку.

Четвертий крок порівнює функцію розподілу досліджуваного тексту з функціями розподілу авторів з бази даних. Алгоритм порівнює відмінності в частотах кожної співпадаючої грами. Якщо певна грама відсутня в розподілі, їй необхідно присвоїти значення «0» для аналізу. Далі нам необхідно скласти суму модулів відхилень грам – це буде середнім відхиленням авторського стилю від аналізованого тексту. Наступним кроком буде порівняння відхилень, автор з найменшим відхиленням буде шуканим автором. Для зручності та аналізу результатів необхідно вивести всі аналізовані варіанти.

Алгоритм можливо покращити, додавши додатковий рівень аналізу, наприклад, більш поглиблений рівень аналізу за допомогою наступної по довжині грами, якщо результати нечіткі.

## 2.4 Висновки

Була проведена формалізація задачі.

Було проаналізовано формули аналізу та визначено, що для роботи алгоритму буде використовуватися метод аналізу частотного розподілу грам, а для аналізу авторства – метод аналізу відстані між функціями щільності тексту та авторів у базі даних.

Було проаналізовано можливі проблеми, що постануть у випадку програмної реалізації алгоритму:

- фільтрація небажаних символів;
- збереження даних для подальшого аналізу;
- відображення результатів.

Було проаналізовано формули аналізу тексту та побудовано алгоритм аналізу тексту згідно поставленої задачі в розділі 1.

# 3 РОЗДІЛ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## 3.1 Загальний опис програми

Програма складається з двох модулів. Перший модуль аналізує твори та дає статистичній характеристики тексту, відповідно до застосованого алгоритму. Другий модуль аналізує базу даних та видає результати співпадіння статистичного аналізу тексту з авторами, що є в базі.

Для розробки була обрана мова програмування Java, як така, що забезпечує велику швидкість обміну даними. Використовується фреймворк Java Spring[15] для швидкого налаштування та запуску віртуальної машини Java(JVM), що призводит до швидшої роботи алгоритму Використовуються бібліотека org.apache.commons для реалізації роботи з українською мовою та бібліотеки org.knowm.xchart і org.apache.poi для виводу даних у таблицю формата xlsx для подальшої обробки.

Було обрані саме ці бібліотеки, як такі, що найбільше відповідають вимогам алгоритму, сформульованого в розділі 2.

## 3.2 Проблеми та метод їх усунення

В розділі два було виявлено три проблеми, що необхідно усунути на етапі розробки програмного додатку.

Перша проблема це необхідність фільтрації тексту. Для виконання можна було б використовувати перевірку виключенням, але більш швидким та надійним варіантом вирішення проблеми буде модифікації аналітичної частини програми за допомогою регулярних виразів, таким чином, щоб програма аналізувала лише необхідні символи.

Друга проблема це збереження даних. Є декілька варіантів усунення цієї проблеми. Для збереження даних можна використати наступні варіанти:

- збереження в файлі текстового формату – просто, швидко і займає найменше місця, але може виникнути проблеми з зчитуванням даних в майбутньому;
- збереження в файлі в форматі xml або подібному до нього – варіант, дуже зручний при зчитуванні даних, але необхідно створити систему зчитування даних, також виникають проблеми з подальшим аналізом даних;
- збереження в табличний файл – варіант зручний для подальшого аналіз даних, та для візуалізацію завдяки вбудованим методам більшості програм, що підтримують роботу з табличними файлами, єдиний недолік – необхідно будувати систему зачитування даних для роботи модулю аналізу авторства;
- збереження в базу даних – цей варіант потребує налаштування та встановлення зв’язків для бази даних, крім того, більшість баз даних не є безкоштовними, і тому цей варіант не є варіантом в даному дослідженні, але його можна використати при подальшому розвитку програмного додатку.

Проаналізувавши варіанти вирішення, було вирішено використати табличний файл формату xlsx, як такого, що дає можливість проводити подальшу роботу з даними аналізу найбільш зручними методами.

Третя проблема – проблема візуалізації даних, у випадках, коли це необхідно відразу по закінченню роботи алгоритму. Для цього була обрана бібліотека org.known.xchar, що дозволяє швидко вивести необхідні дані у потрібному вигляді. Демонстрація результати роботи бібліотеки наявна на рисунку 3.1.

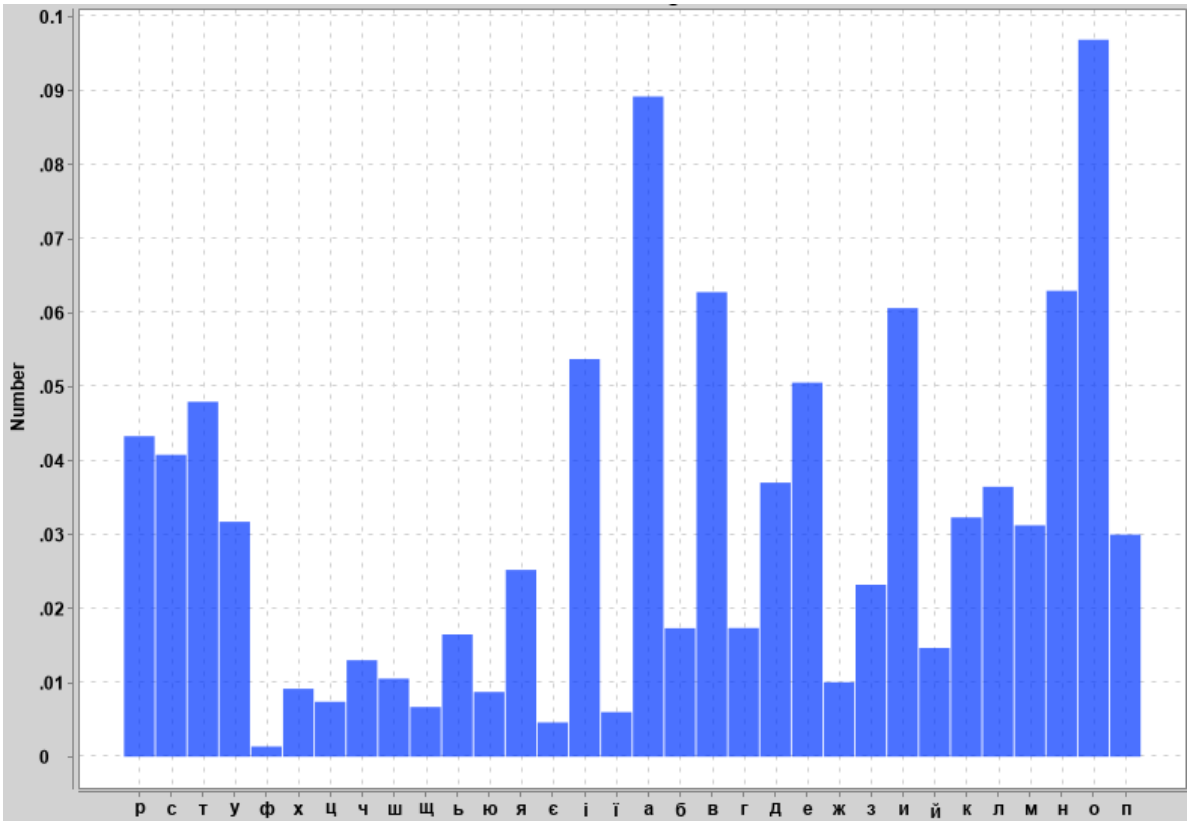



Рисунок 3.1 – приклад виводу результатів аналізу

### 3.3 Модуль аналізу текста

Перед обробкою текст зводиться у нижній регістр, щоб запобігти розподілу літер на великі і малі. Далі будується таблиця, і відбувається прохід по тексту з підрахунком грам, відповідно до обраного алгоритму:

Алгоритм аналізу грамів рахує кількість входжень грамів у тексті прямим проходом, після чого вираховує відносну частоту входження окремої грами і повертає результат в основний алгоритм.

Метод аналізу біграм відрізняється від попереднього наявністю двох режимів – перший відповідає за покроковий аналіз, другий – за перехресний. Суть покроковому – після зчитування біграми алгоритм переходить до наступної. Перехресний метод створює нову біграму з кінцьової літери попередньої біграми та першої літери наступної.

Результати у вигляді таблиці «Грама» - «Частота» виводяться у окремий файл для подальшого аналізу.

Візуалізація алгоритму подана на рисунку 3.2.

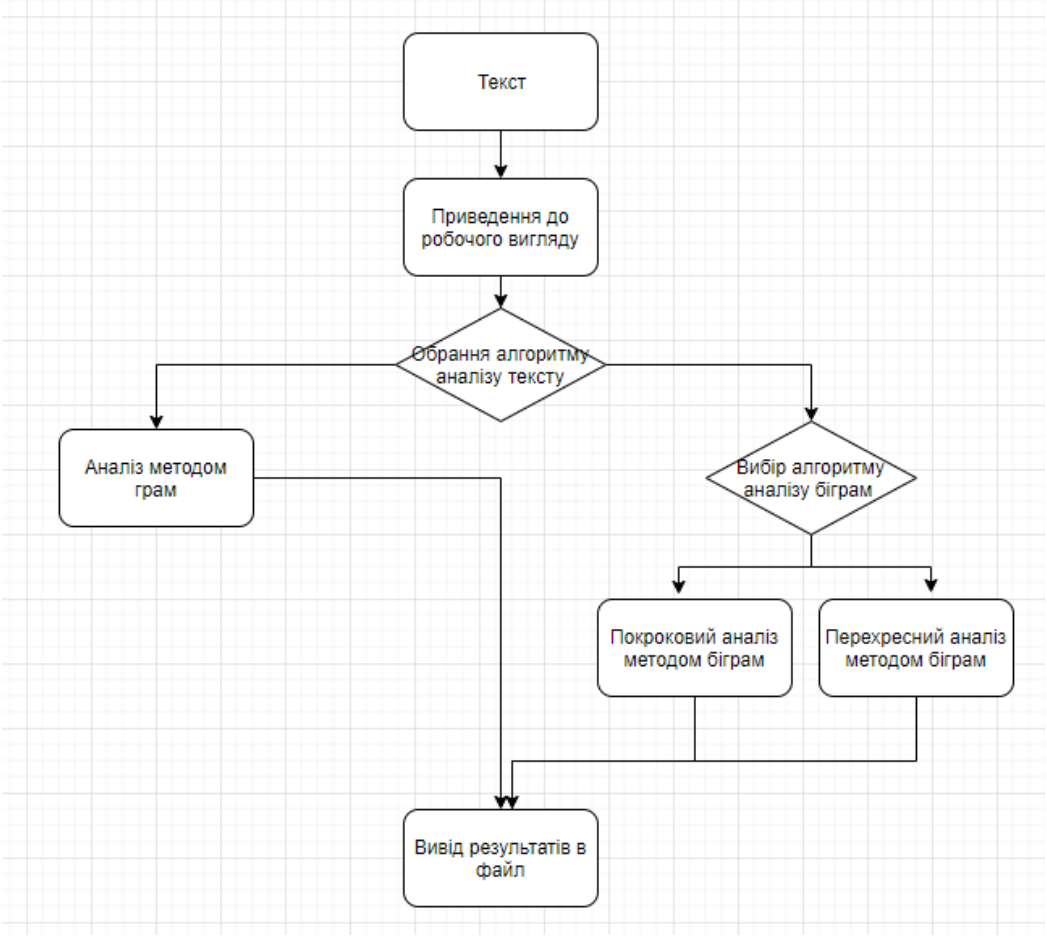



Рисунок 3.2 – модуль аналізу тексту

### 3.4 Модуль аналізу авторства

Для отримання результату аналізу авторства текст проходить аналіз методом грам. Далі програма звертається до бази даних та порівнює функцію щільності тексту з відповідними функціями середньої щільності текстів авторів в базі даних.

В деяких випадках необхідно буде подальше уточнення результату, наприклад, коли є декілька функцій, що мають подібне відхилення з досліджуваним текстом. Тоді текст проходить перевірку методом біграм і проводиться аналіз по відповідній базі.

Візуалізація алгоритму подана на рисунку 3.3.

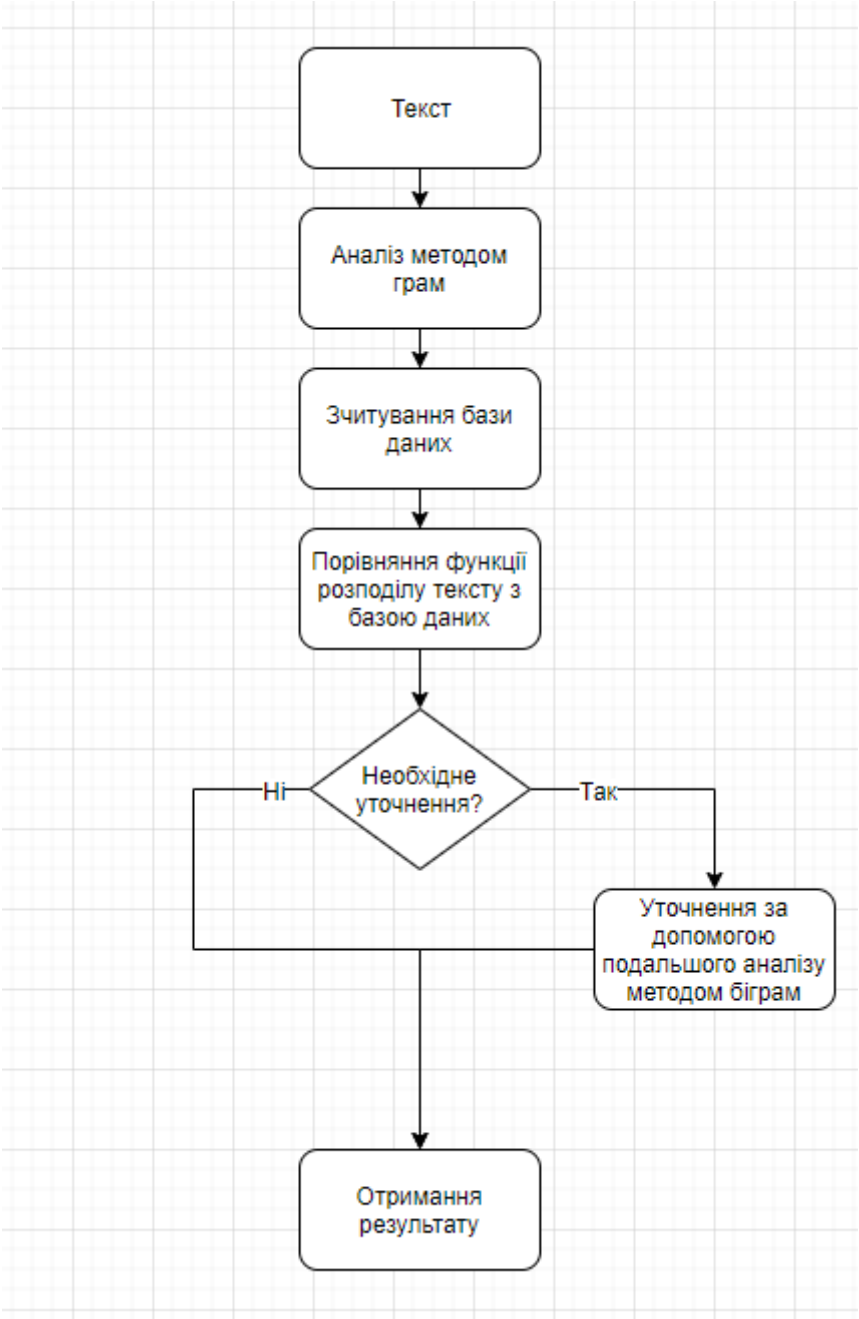



Рисунок 3.3 – модуль пошуку автора

## 3.5 Подальші покращення алгоритму аналізу

У випадках, коли база даних авторів досягне великого розміру, системі необхідно буде метод класифікації, для прискорення роботи алгоритму. Мною був обраний метод k-середніх.

Базу даних авторів можна представити у вигляді багатовимірних векторів, де грама – це напрям, а частота – числове значення. Після чого методом k-середніх база розбивається на кластери. Для прискорення роботи алгоритму можна використовувати не всі грами, а лише ті, що найбільше характеризують автора. Для грам це літери «І», «К», «Л», «С», «Т».

Також можливо розширити проект, перейшовши на роботу з повноцінною базою даних. Це необхідно буде у випадку комерціалізації додатку у вигляді інструменту пошуку автора художнього тексту.

### 3.6 Висновки

Було розроблено програмне забезпечення згідно поставленої задачі в розділі 1 та розробленого алгоритму в розділі 2.

Було проаналізовано проблеми, знайдені в розділі 2, та знайдені методи їх вирішення, а саме:

- для фільтрації тексту в модуль аналізу буде вбудовано фільтратор на регулярних виразах;
- для збереження даних для подальшого аналізу буде використано табличний файл типо xlsx;
- для представлення результатів буде використано бібліотеку xchar.

Програмне забезпечення складається двох модулів. Перший модуль відповідає за аналіз тексту методом статистичного аналізу для створення та наповнення бази авторів Другий модуль відповідає за аналіз тексту та пошук найбільш відповідного автора з бази методом побудови та порівняння функції щільності.

Також було проаналізовані можливі розширення для програмного додатку, а саме:

- метод к-середніх для роботи у випадках з великими базами авторів;
- використання бази даних для розширення можливостей збереження даних і можливості комерціалізації додатку.

# 4 АНАЛІЗ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

## 4.1 Результати аналізу грам

Для аналізу були відібрані 8 українських авторів, а саме: О. Довженко, І. Котляревський, П. Мирний, І. Нечуй-Левицький, П. Куліш, Т. Шевченко, В. Нестайко та О. Вишня. Ці автори були відібрані як представники різних жанрів, що допоможе тестуванню алгоритму. Для аналізу було обрано по кілька творів від кожного, або, у випадку поетів, збірники поезії.

Спочатку було проведено аналіз творів цих авторів, потім було побудовано функцію щільності для творів та знайдено функцію середньої щільності для авторів.

Результати дослідження методом грам представлені на рисунку 4.1. та в таблицях 4.1 та 4.2.

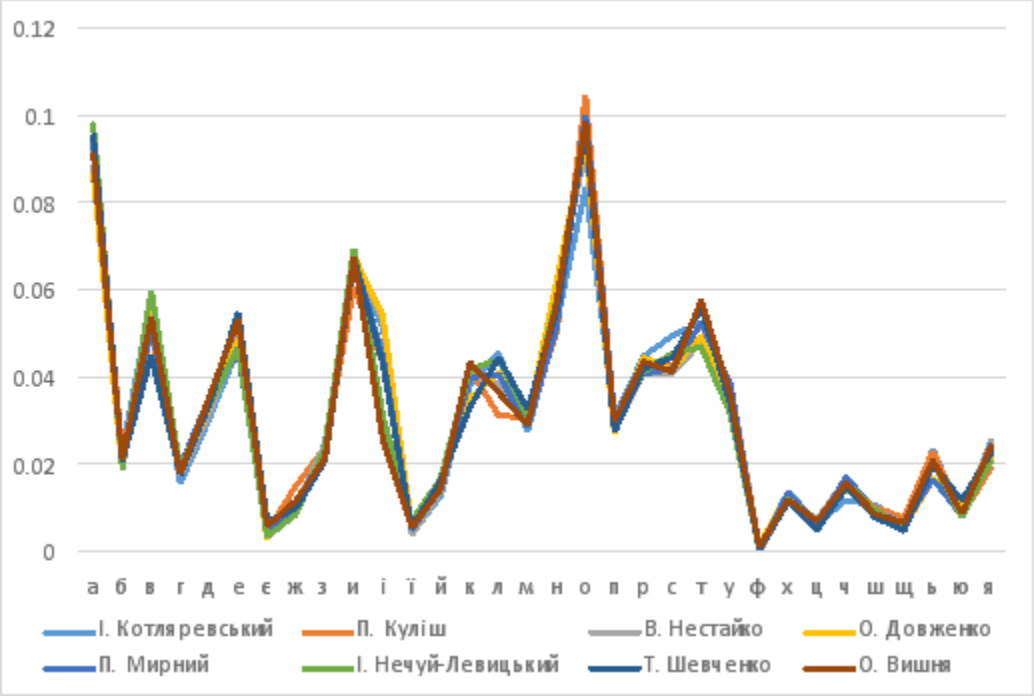



Рисунок 4.1 – аналіз методом грам

Таблиця 4.1 – результати аналізу методом грам, частина 1

| Автор | І. Котляревський | П. Куліш | В. Нестайко | О. Довженко |
|---|---|---|---|---|
| а | 0,085903 | 0,092538 | 0,088513 | 0,086404 |
| б | 0,023061 | 0,02227 | 0,019825 | 0,019141 |
| в | 0,057971 | 0,050064 | 0,058316 | 0,058409 |
| г | 0,015525 | 0,018993 | 0,017297 | 0,017936 |
| д | 0,030308 | 0,034501 | 0,0314 | 0,033315 |
| е | 0,046588 | 0,052356 | 0,049148 | 0,048391 |
| є | 0,003512 | 0,004483 | 0,003726 | 0,002888 |
| ж | 0,009217 | 0,015107 | 0,010323 | 0,010204 |
| з | 0,024768 | 0,023758 | 0,024599 | 0,022315 |
| и | 0,065993 | 0,060538 | 0,066599 | 0,067802 |
| і | 0,05136 | 0,043224 | 0,05303 | 0,053983 |
| ї | 0,004096 | 0,005949 | 0,00387 | 0,005517 |
| й | 0,012807 | 0,015 | 0,013789 | 0,015523 |
| к | 0,038192 | 0,042112 | 0,040298 | 0,035095 |
| л | 0,045584 | 0,031121 | 0,038595 | 0,041059 |
| м | 0,027518 | 0,030712 | 0,029916 | 0,0303 |
| н | 0,052325 | 0,050085 | 0,05942 | 0,061879 |
| о | 0,083382 | 0,1047 | 0,092293 | 0,094714 |
| п | 0,030827 | 0,027989 | 0,027635 | 0,027029 |
| р | 0,044652 | 0,042261 | 0,041132 | 0,044487 |
| с | 0,049503 | 0,042725 | 0,040729 | 0,041739 |
| т | 0,052496 | 0,052425 | 0,048084 | 0,049257 |
| у | 0,038225 | 0,036065 | 0,038343 | 0,035034 |
| ф | 0,000683 | 3,41E-05 | 0,001138 | 0,001288 |
| х | 0,013733 | 0,011651 | 0,011916 | 0,013499 |
| ц | 0,006394 | 0,005894 | 0,007425 | 0,006144 |
| ч | 0,011711 | 0,015167 | 0,016442 | 0,014346 |
| ш | 0,010687 | 0,010159 | 0,009441 | 0,008765 |
| щ | 0,006256 | 0,007683 | 0,00651 | 0,005561 |
| ь | 0,023481 | 0,02305 | 0,016848 | 0,016628 |
| ю | 0,007917 | 0,008297 | 0,007986 | 0,009976 |
| я | 0,025326 | 0,019087 | 0,025414 | 0,021373 |

Таблиця 4.2 – результати аналізу методом грам, частина 2

| Автор | П. Мирний | І. Нечуй-Левицький | Т. Шевченко | О. Вишня |
|---|---|---|---|---|
| а | 0,093437 | 0,097826 | 0,095287 | 0,090905 |
| б | 0,020037 | 0,018852 | 0,020645 | 0,020906 |
| в | 0,051299 | 0,059866 | 0,044834 | 0,053629 |
| г | 0,019486 | 0,018834 | 0,017674 | 0,017804 |
| д | 0,035202 | 0,033677 | 0,033744 | 0,035359 |
| е | 0,045755 | 0,046709 | 0,054853 | 0,053072 |
| є | 0,004495 | 0,003384 | 0,006919 | 0,005758 |
| ж | 0,010723 | 0,008478 | 0,009922 | 0,011638 |
| з | 0,022686 | 0,023742 | 0,020837 | 0,021298 |
| и | 0,067234 | 0,069508 | 0,067153 | 0,067557 |
| і | 0,044704 | 0,03112 | 0,04298 | 0,025576 |
| ї | 0,00486 | 0,006986 | 0,00534 | 0,005469 |
| й | 0,014776 | 0,016816 | 0,016291 | 0,014359 |
| к | 0,039922 | 0,041675 | 0,032918 | 0,043331 |
| л | 0,040567 | 0,043935 | 0,044504 | 0,036405 |
| м | 0,028412 | 0,030576 | 0,032524 | 0,028892 |
| н | 0,051188 | 0,057455 | 0,055151 | 0,057052 |
| о | 0,099942 | 0,097411 | 0,096701 | 0,098466 |
| п | 0,030356 | 0,028913 | 0,027347 | 0,029753 |
| р | 0,040633 | 0,041224 | 0,041909 | 0,043535 |
| с | 0,041962 | 0,045469 | 0,044477 | 0,041197 |
| т | 0,052391 | 0,046829 | 0,055832 | 0,057538 |
| у | 0,038435 | 0,031509 | 0,03389 | 0,036152 |
| ф | 0,000176 | 0,000505 | 0,00044 | 0,001146 |
| х | 0,013464 | 0,012018 | 0,011401 | 0,011894 |
| ц | 0,006079 | 0,006962 | 0,004951 | 0,006972 |
| ч | 0,017107 | 0,015621 | 0,01469 | 0,015693 |
| ш | 0,009182 | 0,009634 | 0,00784 | 0,008578 |
| щ | 0,006312 | 0,005313 | 0,004774 | 0,006558 |
| ь | 0,016583 | 0,020485 | 0,01986 | 0,020861 |
| ю | 0,008267 | 0,008009 | 0,011753 | 0,008685 |
| я | 0,024331 | 0,020657 | 0,022562 | 0,02396 |

Як видно з результатів цього аналізу, кожен автор має свої особливості розподілу літер, так що гіпотеза про відмінність статистичної складової стилю автора підтверджується вже на цьому етапі.

Основними грамами, тобто такими, в яких спостерігаються найбільші відмінності між авторами, є літери «І», «К», «Л», «С», «Т».

### 4.2 Аналіз впливу імен на статистичний розподіл грам

На прикладі повісті «Три мушкетери» О. Дюма було досліджено вплив імен головних героїв на статистичний розподіл грам. Існує гіпотеза, що в окремих творах імена головних героїв впливають на статистику тексту, адже вони трапляються досить часто. Я вирішив перевірити цю гіпотезу, а також перевірити, чи зміниться розподіл, якщо прибрати взагалі всі слова, що починаються з великої літери. Результати представлені в таблиці 4.3 та на рисунку 4.2.

Таблиця 4.3. результати дослідження впливу імен на статистику тексту

| | Без імен головних героїв | Без слів з великої літери | Нормальний текст |
|---|---|---|---|
| а | 0,084916403 | 0,083693936 | 0,089212129 |
| б | 0,017904574 | 0,017314875 | 0,017363463 |
| в | 0,064740972 | 0,066274013 | 0,062784371 |
| г | 0,01792616 | 0,017943965 | 0,017384396 |
| д | 0,03608653 | 0,039081124 | 0,037029633 |
| е | 0,052144415 | 0,048797986 | 0,050568507 |
| є | 0,00477592 | 0,004386913 | 0,004631582 |
| ж | 0,010393698 | 0,009707312 | 0,010079579 |
| з | 0,023954073 | 0,025089559 | 0,023230134 |
| и | 0,06252732 | 0,065050452 | 0,06063762 |
| і | 0,05482001 | 0,053960815 | 0,053726357 |
| ї | 0,006211395 | 0,006282536 | 0,006023674 |
| й | 0,015154508 | 0,015346435 | 0,014696508 |
| к | 0,033334305 | 0,033225939 | 0,032326876 |
| л | 0,037624538 | 0,036872746 | 0,03648745 |
| м | 0,031652749 | 0,031929561 | 0,031259257 |

Продовження таблиці 4.3

| | | | |
|---|---|---|---|
| н | 0,060737834 | 0,059756314 | 0,062969635 |
| о | 0,097560449 | 0,098005417 | 0,096901079 |
| п | 0,03033276 | 0,030932807 | 0,030035682 |
| р | 0,041341661 | 0,040268873 | 0,043308698 |
| с | 0,039772353 | 0,040533878 | 0,040802933 |
| т | 0,045651322 | 0,04433945 | 0,047974821 |
| у | 0,032732053 | 0,033957687 | 0,031747012 |
| ф | 0,00141065 | 0,000765173 | 0,001368018 |
| х | 0,009454703 | 0,009701344 | 0,009168963 |
| ц | 0,007668455 | 0,007483416 | 0,007436699 |
| ч | 0,013451366 | 0,013786246 | 0,013044839 |
| ш | 0,010891257 | 0,010405637 | 0,010562101 |
| щ | 0,006917259 | 0,007063228 | 0,006708206 |
| ь | 0,014951599 | 0,014070351 | 0,016533441 |
| ю | 0,009018664 | 0,009128359 | 0,008746102 |
| я | 0,023940043 | 0,024843653 | 0,025250236 |

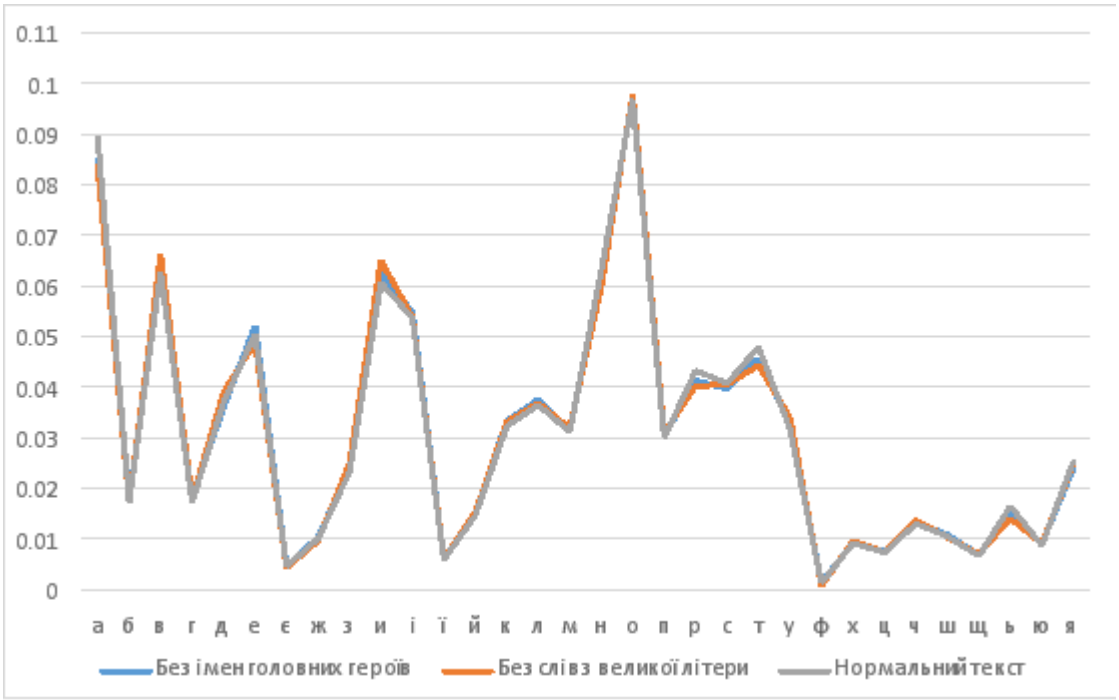



Рисунок 4.2 - результати дослідження впливу імен на статистику тексту

Чітко видно розбіжності на літерах «А», «Е», «И», «Р» і «Т», , отже на статистистичну складову стилю автора в окремому тексті певний вплив створюють імена головних героїв.

### 4.3 Результати визначення автора за методом грам

Програма прораховує відмінності стилю аналізованого твору та стилів авторів у базі даних. Таким чином, чим менше число, там більша ймовірність, що це і є шуканий автор.

Представлено результати аналізу кількох творів.

О. Довженко, «Повість полум’яних літ» - рисунок 4.3.

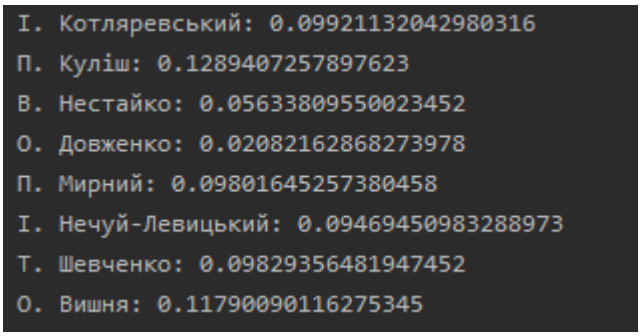



Рисунок 4.3 - О. Довженко, «Повість полум’яних літ»

Як бачимо, в нього відміннсть у стилях мінімальна, отже він і є автором.

І. Нечуй-Левицький, «Кайдашева сім’я» - рисунок 4.4.

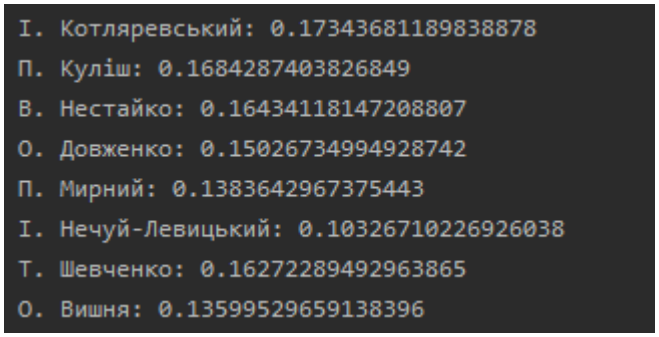



Рисунок 4.4 - І. Нечуй-Левицький, «Кайдашева сім’я»

І знову програма правильно визначила автора.

В. Нестайко, «Одиниця з обманом» - рисунок 4.5

```
І. Котляревський: 0.08904051641022663
П. Куліш: 0.11232563993658642
В. Нестайко: 0.03897968713266633
О. Довженко: 0.05698753828200827
П. Мирний: 0.0773645337317449
І. Нечуй-Левицький: 0.0864377271293984
Т. Шевченко: 0.11776471715309932
О. Вишня: 0.09910871572707555
```

Рисунок 4.5 - В. Нестайко, «Одиниця з обманом»

В. Нестайко, «Тореадори з Васюківки» - рисунок 4.6.

```
І. Котляревський: 0.08459306258576843
П. Куліш: 0.10450839522848836
В. Нестайко: 0.02273823705310165
О. Довженко: 0.05696338439540716
П. Мирний: 0.07926966220588319
І. Нечуй-Левицький: 0.10072734957509269
Т. Шевченко: 0.10774713129715029
О. Вишня: 0.09493208867540402
```

Рисунок 4.6 - В. Нестайко, «Тореадори з Васюківки»

Обидві повісті Нестайка також були розпізнані правильно.

О. Вишня, збірник «Весна-красна» - рисунок 4.7.

```
І. Котляревський: 0.10943034506684049
П. Куліш: 0.08007257678748994
В. Нестайко: 0.06641423184575791
О. Довженко: 0.06546285476239463
П. Мирний: 0.06938907359682706
І. Нечуй-Левицький: 0.10713487739504562
Т. Шевченко: 0.08956459633609769
О. Вишня: 0.06844876544850642
```

Рисунок 4.7 - О. Вишня, збірник «Весна-красна»

Перша помилка, як видно з результатів, тут знайдена близька збіжність одразу з трьома авторами, і результат невірний. Саме для таких випадків і необхідний етап з аналізом біграм.

Після аналізу ще кількох творів, результат правильного аналізу склав приблизно 85,7%. Найбільша кількість похибок спостерігається на збірниках невеликих оповідань або поетичних творів.

Це пов’язано з тим, що на невеликих оповіданнях досить великий розподіл слів, тому алгоритм не може правильно визначити порядок слів.

Необхідно перевірити роботу програми на уривках з текстів. Результати представлені на рисунках 4.8, 4.9, 4.10 та 4.11.

```
I. Котляревський: 0.12459407834563724
П. Куліш: 0.0403873983832662
В. Нестайко: 0.09743501934280882
О. Довженко: 0.10071191724123471
П. Мирний: 0.08267117852432024
І. Нечуй-Левицький: 0.10641759931184848
Т. Шевченко: 0.10638528525639174
О. Вишня: 0.09082585854091292
```

Рисунок 4.8 – Аналіз уривку з повісті «Чорна Рада» П. Куліша

```
I. Котляревський: 0.10287937556355732
П. Куліш: 0.11216442455211104
В. Нестайко: 0.049813772029387415
О. Довженко: 0.08606118081095393
П. Мирний: 0.08738709659120253
І. Нечуй-Левицький: 0.09679716461211242
Т. Шевченко: 0.11171096585651168
О. Вишня: 0.10717909840470216
```

Рисунок 4.9 – Уривок з «Тереадорів з Васюківки» В Нестайка

```
I. Котляревський: 0.15779981862001952
П. Куліш: 0.11688374997426523
В. Нестайко: 0.12214189856280434
О. Довженко: 0.11429480448297061
П. Мирний: 0.1013312950921136
І. Нечуй-Левицький: 0.13506334292222874
Т. Шевченко: 0.09096546730175661
О. Вишня: 0.12173613013782175
```

Рисунок 4.10 – Кілька віршів Т.Шевченка з збірника «Кобзар»

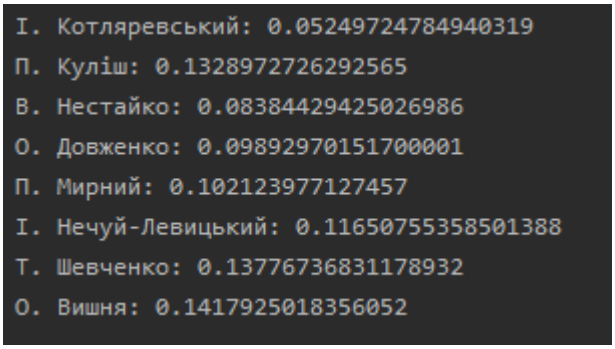



Рисунок 4.11 – Уривок з «Енеїди» І Котляревського

Отже, алгоритм зберігає свою стійкість і можливість аналізу не тільки на великих творах, а й на відносно малих уривках.

### 4.4 Висновки

Було проведено тестування програмного додатку, розробленого в розділі 3. Було проведено порівняльний аналіз частотного розподілу авторів художніх творів української мови. Було підтверджено, що кожен автор має свої особливості розподілу літер, так що гіпотеза про відмінність статистичної складової стилю автора підтверджується вже на цьому етапі.

Основними грамами, тобто такими, в яких спостерігаються найбільші відмінності між авторами, є літери «І», «К», «Л», «С», «Т».

Далі проведено дослідження впливу імен героїв та слів з великої літери на розподіл грам в тексті. На прикладі повісті «Три мушкетери» А. Дюма було доведено, що імена героїв, як одні з найбільш статистично ймовірних слів у тексті, мають значний вплив на частотний розподіл.

Було проведено тестування модулю визначення автора з розділу 3. На досліджуваних творах точність роботи алгоритму становить 85,7%, що є достатнім для подальшої експлуатації додатку.

Також було проведено аналіз швидкості роботи програмного застосунку. Середній час роботи алгоритму при аналізі повісті обсягом 200-300 сторінок займає усього 550 мілісекунд, що є значно вищим за можливих конкурентів. При

аналізі більших творів часто росте пропорційно, так, при аналізі твору на 600 сторінок час роботи склав 1 секунду 153 мілісекунди.

Також було протестовано можливість твору оперувати уривками з текстів. Алгоритм показує свою здатність продовжувати роботу на уривках не менше 5-6-сторінок.

# 5 РОЗРОБКА СТАРТАПУ НА ОСНОВІ ДОСЛІДЖЕННЯ

## 5.1 Опис ідеї проекту

Системи пошуку автора завжди залишаються актуальними, адже спроби видати чужий твір за свій зустрічаються не лише в нуковій галузі, а й в художній. Ринок потребує більшу кількість систем з різними методами аналізу, які будуть незалежними від таких методів обходу перевірки на антиплагіат, як заміна літер та переклад на іншу мову.

Основною метою розробки є створення простого, швидкого і надійного програмного забезпезпечення для аналізу авторства тексту, що буде відповідати вимогам ринку та буде стійким до сучасних методів обходу перевірки.

Таблиця 5.1 – Опис ідеї стартап-проекту

| Зміст ідеї | Напрямки застосування | Вигоди для користувача |
|---|---|---|
| Визначення авторства тексту за допомогою алгоритму на основі статистичного аналізу тексту методом грам | 1. Аналіз авторства тексту | Простий, швидкий та надійни спосіб перевірки авторства тексту, що є стійким до сучасних методів обходу перевірки та не потребує завантаження великих баз даних для перевірки |

Далі були проаналізовані найбільш популярні конкуренти для розроблюваного програмного забезпечення:

Advego Plagiatus. Одна з найстаріших програм для Windows з перевірки унікальності. Її безглуздо використовувати для роботи з великими клієнтами:

програма видає занадто низький відсоток унікальності, через спроби прискорити алгоритм в останньому апдейті.

Переваги Advego:

- немає обмежень на кількість перевіряється тексту;
- потужна підтримка;
- багато функцій і налаштувань;
- швидке оновлення.

Недоліки Advego:

- трохи повільніше основного конкурента - програми Etxt Антиплагіат;
- маленька довжина шингли за замовчуванням;
- зайва підозрілість;
- часта поява капчті.

Etxt Антиплагіат. Гідний конкурент Advego Plagiatus. Має більш зручний інтерфейс, працює трохи швидше і трохи більше коректніше шукає плагіат.

Переваги Etxt:

- немає обмежень на кількість перевіряється тексту;
- багато функцій і налаштувань;
- швидке оновлення;
- приємний інтерфейс;
- швидше Advego Plagiatus.

Недоліки Etxt:

- маленька довжина шингли за замовчуванням;
- зайва підозрілість;
- часта поява капчті.

Content-watch.ru. Свого часу Content-watch не було гідної альтернативи: на моїй пам'яті - це один з перших сервісів, який в принципі визначав рерайт, в той час як конкурентиеще були не в змозі зробити це.

Втім, і зараз Content-watch може похвалитися цікавими алгоритмами роботи з перевірки унікальності текстів. Основна його перевага, на мій погляд - відсутність паніки і помилкових спрацьовувань. Тобто, якщо вже сервіс визначив рерайт, то рерайт дійсно має місце бути. Тому його можна рекомендувати саме копирайтерам, як надійний засіб позбавитися від необгрунтованих претензій замовника.

Плюс для професійних веб-розробників є можливість реалізації платних програмних перевірок, при цьому вартість API найдешевша на ринку на сьогоднішній день.

Переваги Content-watch.ru:

- відсутність паніки і помилкових спрацьовувань;
- швидка і точна перевірка;
- стабільна робота;
- найдешевша реалізація API.

Недоліки Content-watch.ru:

- обмеження за обсягом перевіряється тексту;
- маленький ліміт безкоштовних перевірок;
- невдалий застарілий інтерфейс.

Таблиця 5.2 – Визначення характеристик ідеї проекту

| Техніко-економічні характеристики ідеї | (потенційні) товари/концепції конкурентів | | | | W | N | S |
|---|---|---|---|---|---|---|---|
| | Мій проект | Advego Plagiatus | Etxt Антиплагіат | Content-watch.ru | | | |
| Швидка робота | Так | Ні | Так | Так | | + | |
| Використання в оффлайн режимі | Так | Так | Так | Ні | + | | |
| Зручний інтерфейс | Так | Ні | Так | Ні | + | | |
| Точність перевірки | Так | Ні | Ні | Так | + | | |
| Відсутність помилкових спрацьовувань | Так | Ні | Ні | Ні | + | | |

## 5.2 Технологічний аудит ідеї проекту

Для реалізації задачі перевірки авторства тексту використовуються наступні підходи:

- проектування інтерфейсу за допомогою засобів фреймворку Vaadin;
- використання Amazon RDS для створення бази даних.

Vaadin дозволяє швидко та дешево створити сучасний зручний інтерфейс, що буде зрозумілий будь-якому потенційному клієнту. Крім того, цей фреймворк

дозволяє налаштувати відображення бази даних, що довзволить зробити детальне відображення результатів пошуку.

Amazon Relational Database Service (Amazon RDS) дозволяє просто налаштовувати, використовувати і масштабувати реляційні бази даних в хмарі. Сервіс забезпечує економічне і масштабується використання ресурсів при одночасній автоматизації трудомістких завдань адміністрування, таких як виділення апаратного забезпечення, налаштування бази даних, установка виправлень і резервне копіювання. Це дозволяє зосередити увагу на додатках, щоб забезпечити для них високу продуктивність, високу доступність, безпеку і сумісність.

Ядро програми буде розроблятися мовою програмування Java за допомогою фреймворку Spring.

Таблиця 5.3 – Технологічна здійсненність ідеї проекту

| Технології її реалізації | Наявність технологій | Доступність технологій |
|---|---|---|
| Vaadin | + | + |
| Amazon RDS | + | + |
| Обрана технологія реалізації ідеї проекту: Java. | | |

### 5.3 Аналіз ринкових можливостей запуску стартап-проекту

Для визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, що дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів, було проведено аналіз попиту (таблиця 5.4).

Таблиця 5.4 – Попередня характеристика потенційного ринку стартап-проекту

| Показник стану ринку | Характеристика |
|---|---|
| Кількість головних гравців, од | 3 |
| Можливі річні обсяги випуску в натуральних показниках | 600 Basic ліцензій,<br>400 Advanced ліцензій |
| Ціна одиниці продукції | Залежить від типу ліцензії:<br>Basic – $30<br>Advanced – $60 |
| Річні обсяги випуску в вартісних показниках | $42 000 |
| Динаміка ринку (якісна оцінка) | Зростає |
| Наявність обмежень для входу (вказати характер обмежень) | Наявність великої кількості більш знайомих користувачам конкурентів |
| Специфічні вимоги до стандартизації та сертифікації | Немає |
| Середня норма рентабельності в галузі (або по ринку), % | 76% |

Потенційні групи клієнтів, їх характеристики, та орієнтовний перелік вимог до товару для кожної групи наведено у таблиці 5.5.

Таблиця 5.5 – Характеристика потенційних клієнтів стартап-проекту

| Потреба, що формує ринок | Цільова аудиторія (цільові сегменти ринку) | Відмінності у поведінці різних потенційних цільових груп клієнтів | Вимоги споживачів до товару |
|---|---|---|---|
| Аналіз авторства тексту | Дрібні автори | Необхідне дешеве та просте рішення з базовим функціоналом | - зручність і простота використання;<br>- можливість гнучкої конфігурації; |
| | Відомі автори та великі видавництва | Необхідне рішення, що дозволить паралельно аналізувати велику кількість творів | - зручність і простота використання;<br>- можливість гнучкої конфігурації;<br>- можливість одночасного аналізу кількох творів. |

Результати аналізу ринкового середовища подані у таблиці факторів, що сприяють ринковому впровадженню проекту (таблиця 5.7), та факторів, що йому перешкоджають (таблиця 5.6).

Таблиця 5.6 – Фактори загроз

| Фактор | Зміст загрози | Можлива реакція компанії |
|---|---|---|
| Поява конкурентів | Поява якіснішого продукту.<br>Поява більш дешевшого продукту. | Розробка удосконалень ПЗ, зменшення вартості продукту, додавання нового функціоналу. |
| Економічний спад | Відсутність попиту на товар через економічну кризу | Зменшення вартості продукту, пошук нових ринків, зміна цільової аудиторії. |

Таблиця 5.7 – Фактори можливостей

| Фактор | Зміст можливості | Можлива реакція компанії |
|---|---|---|
| Розширення ринку | Вихід видавців газет та журналів на ринок | Поширення діяльності на новий ринок. |
| Зменшення кількості конкурентів | Закриття компаній-конкурентів. Погіршення якості конкурентного продукту. | Переманювання та початок роботи з новими партнерами. |
| Покращення репутації компанії | Стабільна робота продукту, задоволення потреб партнерів та клієнтів. | Збільшення обсягів продажів, розробка нового функціоналу, робота з новими партнерами. |

Загальні риси конкуренції на ринку наведені у таблиці 5.8

Таблиця 5.8 – Ступеневий аналіз конкуренції на ринку

| Особливості конкурентного середовища | В чому проявляється дана характеристика | Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною) |
|---|---|---|
| Монополістична конкуренція | На даний момент на ринку існує доволі велика кількість продавців, що надають схожі продукти | Встановлення нижчої ціни, покращення якості власного продукту |
| Національний рівень | Потенційні клієнти знаходяться на всій території держави | Проведення презентацій з демонстрацією роботи ПЗ у обласних центрах та на національних конференціях |
| Внутрішньогалузева | Проект призначений в першу чергу для авторів та видавництв | Розвивати продукт в межах зазначеної сфери |
| Товарно-видова | Продукт конкурує з подібними товарами того ж виду | Впровадження унікального функціоналу |

| | | |
|---|---|---|
| Цінова | Ціна використовується як засіб досягнення кращих економічних умов збуту | Створення спрощеної та дешевшої версії продукту |

Продовження таблиці 5.8

| | | |
|---|---|---|
| Марочна | Важливу роль відіграє ставлення до бренду компанії | Реклама, співпраця з відомими партнерами |

Більш детальний аналіз умов конкуренції в галузі за моделлю 5 сил М. Портера наведено у таблиці 5.9.

Таблиця 5.9 – Аналіз конкуренції в галузі за М. Портером

| | Прямі конкуренти в галузі | Потенційні конкуренти | Постачальники | Клієнти | Товари-замінники |
|---|---|---|---|---|---|
| Склад о | Advego Plagiatus; Etxt Антиплагіат; Content-watch.ru | Конкурентоспроможна ціна, якісніший продукт | Відсутні | Мають сильний вплив на ринок | Відсутні |

| | | | | | |
|---|---|---|---|---|---|
| Висновк | Немає агресивної боротьби за ринок збуту | Вихід на ринок можливий, сторки виходу – 3-6 місяців | – | Клієнти формують вимоги до функціоналу продуку та примушують ринок розвиватися у заданому напрямку | – |

Зважаючи на результати аналізу конкурентної ситуації можна зробити висновок, що вихід на ринок не є складним, основними характеристиками для забезпечення конкурентоспроможності продукту є реалізація функціоналу необхідного клієнтам та невисока ціна.

На основі аналізу конкуренції (таблиця 5.9), а також із урахуванням характеристик ідеї проекту (таблиця 5.2), вимог споживачів до товару (таблиця 5.5) та факторів маркетингового середовища (таблиці 5.6 та 5.7) визначено перелік факторів конкурентоспроможності, що подані у таблиці 5.10.

Таблиця 5.10 – Обґрунтування факторів конкурентоспроможності

| Фактор конкурентоспроможності | Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим) |
|---|---|
| Гнучка система ліцензування | Клієнт може придбати саме той функціонал, що йому необхідний |
| Низька конкуренція | Мало конкуруючих продуктів |
| Унікальний функціонал | Наявність можливостей, що не реалізовані у конкурентів |

| | |
|---|---|
| Простота та безпека інтеграції | Не потребує внесення змін до існуючої інформаційної системи підприємтва |
| Відомість бренду | Клієнти надають велике значення бренду та репутації компанії |
| Якість сервісного обслуговування | Терміни реагування на відгуки клієнтів та якість надання послуг підпримки |

Аналіз сильних та слабких сторін стартап-проекту за визначеними факторами конкурентоспроможності (таблиця 5.10) наведено у таблиці 5.11.

Таблиця 5.11 – Порівняльний аналіз сильних та слабких сторін "Системи автоматизації складських та торгових процесів"

| Фактор конкурентоспроможності | Бали 1-20 | Рейтинг товарів-конкурентів | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | -3 | -2 | -1 | 0 | 1 | 2 | 3 |
| Гнучка система ліцензування | | | | | | | + | |
| Низька конкуренція | | | | | + | | | |
| Унікальний функціонал | | | | | | + | | |
| Простота та безпека інтеграції | | | | | | | + | |
| Відомість бренду | | | + | | | | | |
| Якість сервісного обслуговування | | | | | | + | | |

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) на основі виділених

ринкових загроз та можливостей, та сильних і слабких сторін, що наведений у тиблиці 5.12 .

Таблиця 5.12 – SWOT-аналіз стартап-проекту

| Сильні сторони: | Слабкі сторони: |
|---|---|
| - наявність унікального функціоналу<br>- гнучка система ліцензування<br>- простота та безпека інтеграції<br>- якість сервісного обслуговування | - невідомість бренду<br>- відсутність партнерів<br>- брак коштів на маркетинг |

Продовження таблиці 5.12

| Можливості: | Загрози: |
|---|---|
| - збільшення клієнтської бази<br>- здобуття репутації<br>- впровадження нового функціоналу<br>- пошерення на нові ринки | - зниження попиту<br>- поява нових конкурентів<br>- закриття ринку |

На основі SWOT-аналізу було розроблено альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок.

Визначені альтернативи наведені у таблиці 5.13.

Таблиця 5.13 – Альтернативи ринкового впровадження стартап-проекту

| Альтернатива ринкової поведінки | Ймовірність отримання ресурсів | Строки реалізації |
|---|---|---|

| | | |
|---|---|---|
| Нарощення клієнтської бази шляхом проведення презентацій та розміщення реклами | Висока | 2-4 місяці |
| Вихід на нові ринки | Середня | 3-6 місяці |

Більш перспективною ринковою поведінкою буде нарощення клієнтської бази та популяризація бренду на існуючно ринку.

### 5.4 Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів, що наведено у таблиці 5.14.

Таблиця 5.14 – Вибір цільових груп потенційних споживачів

| Опис цільової групи потенційних клієнтів | Готовність споживачів сприйняти продукт | Орієнтовний попит у сегменті | Інтенсивність конкуренції в сегменті | Простота входу у сегмент |
|---|---|---|---|---|
| Дрібні автори | Продукт потрібний | Середній | Значна | Середня складність |

| | | | | |
|---|---|---|---|---|
| Відомі автори та великі видавництва | Продукт потрібний | Високий | Незначна | Просто |

Зважаючи на те, що компанія працюватиме з всім ринком, доцільно пропонувати стандартизовану програму та використовувати масовий маркетинг.

Для роботи в обраних сегментах ринку необхідно сформувати базову стратегію розвитку, яка визначається у таблиці 5.15.

Таблиця 5.15 – Визначення базової стратегії розвитку

| Обрана альтернатива розвитку проекту | Стратегія охоплення ринку | Ключові конкурентоспроможні позиції відповідно до обраної альтернативи | Базова стратегія розвитку |
|---|---|---|---|

| | | | |
|---|---|---|---|
| Нарощення клієнтської бази шляхом проведення презентацій та розміщення реклами | Стратегія диференціації | Розробка унікального функціоналу, гнучка система ліцензування | Стратегія диференціації |

Вибір стратегії конкурентної поведінки наведено у таблиці 5.16.

Таблиця 5.16 – Визначення базової стратегії конкурентної поведінки

| Чи є проект «першопрохідцем» на ринку? | Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів? | Чи буде компанія копіювати основні характеристики товару конкурента, і які? | Стратегія конкурентної поведінки |
|---|---|---|---|
| Ні | Обидва варіанти | Ні | Стратегія лідера |

На основі вимог споживачів з обраних сегментів до постачальника (стартап-компанії) та до продукту, а також в залежності від обраної базової стратегії розвитку та стратегії конкурентної поведінки було розроблено стратегію позиціонування (таблиця 5.17). що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку/проект.

Таблиця 5.17 – Визначення стратегії позиціонування

| | | | |
|---|---|---|---|
| Вимоги до товару цільової аудиторії | Базова стратегія розвитку | Ключові конкурентоспроможні позиції власного стартап-проекту | Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових) |

Продовження таблиці 5.17

| | | | |
|---|---|---|---|
| Зручність і простота використання, можливість гнучкої конфігурації, можливість розділення обов'язків між персоналом, можливість гнукого налаштування параметрів відображення інформації | Стратегія диференціації | - наявність унікального функціоналу<br>- гнучка система ліцензування<br>- простота та безпека інтеграції<br>- якість сервісного обслуговування | - гнучкість<br>- універсальність<br>- простота |

Робота стартап-компанії на ринку повинна бути спланована за стратегією диференціації, що передбачає надання товару важливих з точки зору споживача відмітних властивостей, які роблять товар відмінним від товарів конкурентів.

У конкурентній поведінці компанія буде дотримуватися стратегії "лідера", розширюючи первинний попит, оскільки ринок є доволі молодим та ненасиченим.

### 5.5 Розроблення маркетингової програми стартап-проекту

Першим кроком є формування маркетингової концепції товару, який отримає споживач. Для цього у таблиці 5.18 підсумувано результати попереднього аналізу конкурентоспроможності товару.

Таблиця 5.18 – Визначення ключових переваг концепції потенційного товару

| Потреба | Вигода, яку пропонує товар | Ключові переваги перед конкурентами |
|---|---|---|
| Аналіз авторства тексту | Простий, надійний та зручний спосіб перевірки авторства тексту, що не потребує завантаження великих баз даних або постійного знаходження в мережі Інтернет | - робота в оффлайн режимі<br>- гнучка система ліцензування<br>- простота та зручність роботи<br>- якість сервісного обслуговування |

Трирівнева маркетингова модель товару, що уточнює ідею продукту, його фізичні складові, особливості процесу його надання, наведена у таблиці 5.19.

Таблиця 5.19 – Визначення стратегії позиціонування

| Рівні товару | Сутність та складові |
|---|---|
| І. Товар за задумом | Визначення авторства тексту за допомогою алгоритму на основі статистичного аналізу тексту методом грам |

Продовження таблиці 5.19

| | |
|---|---|
| II. Товар у реальному виконанні | 1. Робота в офлайн режимі<br>2. Простота та зручність роботи<br>3. Гнучка система ліцензування |
| | Якість: проведення публічного тестування |
| | Марка: DSHCH Software – DSHCH |
| III. Товар із підкріпленням | Після продажу: додавання нового функціоналу. |
| Для захисту ПЗ від копіювання буде застосовано механізм ліцензування. | |

Визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар, що передбачає аналіз ціни на товари-аналоги або товари субститути, а також аналіз рівня доходів цільової групи споживачів подано у таблиці 5.20.

Таблиця 5.20 – Визначення меж встановлення ціни

| Рівень цін на товари-аналоги | Рівень доходів цільової групи споживачів | Верхня та нижня межі встановлення ціни на товар/послугу |
|---|---|---|
| $25-$100 | Від $4000 | $30-$60 |

Визначення оптимальної системи збуту наведено у таблиці 5.21.

Таблиця 5.21 – Формування системи збуту

| Специфіка закупівельної поведінки цільових клієнтів | Функції збуту, які має виконувати постачальник товару | Глибина каналу збуту | Оптимальна система збуту |
|---|---|---|---|
| Надання переваги оптовим закупівлям | Продаж товару та пошук нових ринків збуту | Однорівневий та дворівневий канали | Вертикальна |

Останньою складової маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування та визначену специфіку поведінки клієнтів (таблиця 5.22).

Таблиця 5.22 – Формування системи збуту

| Специфіка поведінки цільових клієнтів | Канали комунікацій, якими користуються цільові клієнти | Ключові позиції, обрані для позиціонування | Завдання рекламного повідомлення |
|---|---|---|---|

| | | | |
|---|---|---|---|
| Надання переваги оптовим закупівлям | Будь-які | Універсальний рішення;<br>Гнучка система ліцензування;<br>Наявність унікального функціоналу | Зацікавити якомого ширше коло потенційних клієнтів |

### 5.6 Економічне обґрунтування стартап-проекту

Для програмної реалізації продукту необхідно залучення трьох спеціалістів та сумарно вимагає затрати 700-900 робочих годин. Середня вартість годин роботи спеціаліста на терирорії України складає $14. Відповідно загальна вартість розробки оцінюється у $9 800-12 600.

З метою популяризації бренду компанії та для успішної реалізації товару слід провести рекламну кампанію, що включатиме організацію презентацій, виступи на конференціях, рекламу на спеціалізованих ресурсах в інтернеті. Також передбачається безкоштовне надання демонстраційних ліцензій, що надаватимуть доступ до обмеженого функціоналу ПЗ з метою зацікавлення широкого кола потенційних клієнтів. Сумарні маркетингові витрати оцінюються у $5 500-7 000.

Для фінансового забезпечення розробки та просування продукту на ринку необхідне залучення партнера, на умовах поділу прибутку від реалізації у рівних долях. Зважаючи на можливі річні обсяги реалізації продукту описані раніше (600 Basic та 400 Advanced ліцензій сумарною вартістю $42 000) та сумарні витрати на виробництво та маркетинг ($15 300 – 19 600) партнер може розраховувати на повернення вкладених коштів через 12-15 місяців з моменту початку розробки.

### 5.7 Висновки

Проект розробки "Система пошуку автора на основі статистичного аналізу" користується попитом з боку потенційних клієнтів, оскільки на даний момент на ринку немає рішення, що реалізовує весь описаний функціонал, а існуючі конкуренти мають вищу ціну та не здатні задовольнити потреби ринку.

Зважаючи на середній рівень конкуренції, проект має високу ймовірність успішного виходу на ринок та комерціалізації. Найбільш перспективним варіантом впровадження системи є нарощення клієнтської бази на локальному ринку з подальшим розширенням на території СНД. Проект передбачає залучення партнера для фінансового забезпечення розробки з подальшим поділом прибутку в рівних долях.

# ВИСНОВКИ

Задача аналізу авторства та визначення стилю за допомогою статистичного аналізу дозволить з високою точністю визначати автора.

Було вивчено предметну область та побудовано алгоритм на основі методів статистичного аналізу текстів, а саме, методі аналізу грам.

Було розроблене програмне забезпечення, що проводить статистичний аналіз і може з високою точністю визначити авторство художніх творів.

Було проведено аналіз проблем при розробці та знайдено вирішення для кожної з них а саме:

- для фільтрації тексту в модуль аналізу буде вбудовано фільтратор на регулярних виразах;
- для збереження даних для подальшого аналізу буде використано табличний файл типо xlsx;
- для представлення результатів буде використано бібліотеку xchar.

Програмний додаток можливо комерціалізувати та розвивати в напрямку інстурменту для пошуку автора.

Аналіз текстів підтвердив гіпотезу про значну відмінність статистичної складової стилю автора. Найбільш відмінні грами при аналізі такстів українських авторів - літери «І», «К», «Л», «С», «Т».

На прикладі твору «Три Мушкитери» була досліджена та підтверджена гіпотеза про вплив імен головних героїв на статистику тексту.

Була визначена точність 85,7% при аналізі такту на авторство методом грам. Для збільшення точності необхідно проводити подальший аналіз текстів зі слабкою визначеністю методом біграм.

Також була підтверджена значна швидкість алгоритму і можливість роботи не тільки з повноцінними текстами, а й з уривками з них.

Було побудовано стартап на основі дослідженого методу визначення авторства.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1) О.О. Архипова В.М. Журавльов: Частотний аналіз використання букв української мови: https://cyberleninka.ru/article/v/chastotniy-analiz-vikoristannya-bukv-ukrayinskoyi-movi

2) Baudouin C. Elements de cryptographie / C. Baudouin, Ed. A. Pedone. – Paris, 1939. – 214 p.

3) Борисов Л.А., Орлов Ю. Н., Осминин К.П.: Идентификация автора текста по распределению частот буквосочетаний: https://library.keldysh.ru/preprint.asp?id=2013-27

4) Орлов Ю.Н., Осминин К.П. Методы статистического анализа литературных текстов. – М.: Эдиториал УРСС/Книжный дом «ЛИБРОКОМ», 2012. – 326 с.

5) Орлов Ю.Н., Осминин К.П. Определение жанра и автора литературного произведения статистическими методами // Прикладная информатика, 2010. Т. 26. № 2. С. 95-108.

6) Романов А.С. Методика и программный комплекс для идентификации автора неизвестного текста: Автореф. дис. канд. техн. наук. Томск, 2010. 26 с.

7) Лазутина Г. В. Основы творческой деятельности журналиста. Учебник для студентов вузов, гриф Минобрнауки. — М., Аспект Пресс, 2005

8) Костенко Н., Іванов В. Досвід контент-аналізу. Моделі та практики. / Наталія Костенко, Валерій Іванов. — К.: Центр вільної преси, 2003

9) Статистичний аналіз [Електронний ресурс] https://stud.com.ua/49878/marketing/statistichniy_analiz

10) Королюк В.С., Портенко Н.И., Скороход А.В., Турбин А.Ф. Справочник по теории вероятностей и математической статистике. – М.: Наука, 1985. – 640 с.

11) Абрамовиц М., Стиган И.М. Справочник по специальным функциям. – М.: Наука, 1979.

12) Покровский Н.Б Расчёт и измерение разборчивости речи / Н. Б. Покровский. – М. : Связьиздат, 1962. – 392 с.

13) Електронна бібліотека українських творів УкрЛіб [Електронний ресурс] https://www.ukrlib.com.ua/books/

14) Електронна бібліотека художніх творів JavaLibre [Електронний ресурс] https://javalibre.com.ua/

15) Фреймворк Java Spring [Електронний ресурс] https://spring.io/

# ДОДАТОК А ОПИС ПРОГРАМИ

## Опис модулю аналізу тексту LetterChart.java

```
package com.dip.mdautor;

import org.apache.poi.ss.usermodel.*;
import
org.apache.poi.xssf.usermodel.XSSFWorkbook;
import org.knowm.xchart.*;
import org.knowm.xchart.style.Styler;

import javax.swing.*;
import java.io.*;
import java.util.*;

public class LetterChart extends JFrame{

   static CharParser charParser = new
CharParser();
   private static List<String>
tagsFinderMainTestInputArray = new
ArrayList<>();
   private static String wholeDoc = "";

   public static void main(String[] args) throws
IOException {
      BufferedReader reader = new
BufferedReader(new InputStreamReader(
            new
FileInputStream("C:\\Users\\DMYTROShCHE
RBAKOV\\Documents\\Projects\\study\\mdauto
r\\src\\main\\resources\\Try-Mushketery-
Aleksandr-Dyuma (1).txt"), "UTF8"));
      String line = reader.readLine();
      while(line != null) {

tagsFinderMainTestInputArray.add(line);
         wholeDoc = wholeDoc + line;
         line = reader.readLine();
      }
      reader.close();
//       for(int i=0;
i<tagsFinderMainTestInputArray.size(); i++) {
//
charParser.charParse(tagsFinderMainTestInput
Array.get(i));
//       }
      HashMap<Character, Double> countMap =
charParser.charParse(wholeDoc);
      ArrayList<String> letters = new
ArrayList<>();
      ArrayList<Double> counts = new
ArrayList<>();
      for (Map.Entry<Character, Double> e:
countMap.entrySet()) {
         letters.add(String.valueOf(e.getKey()));
         counts.add(e.getValue());
      }
      HashMap<String, Double> output = new
HashMap<>();
      for (Map.Entry<Character, Double> e:
countMap.entrySet()) {
         output.put(String.valueOf(e.getKey()),
e.getValue());
      }
      outputToEXCEL(output);
      wholeDoc = "";
      reader = new BufferedReader(new
InputStreamReader(
            new
FileInputStream("C:\\Users\\DMYTROShCHE
RBAKOV\\Documents\\Projects\\study\\mdauto
r\\src\\main\\resources\\PKulish\\Chorna_rada.tx
t"), "UTF8"));
      line = reader.readLine();
      while(line != null) {

tagsFinderMainTestInputArray.add(line);
         wholeDoc = wholeDoc + line;
         line = reader.readLine();
      }
      reader.close();
      HashMap<Character, Double> countMap2
= charParser.charParse(wholeDoc);
      ArrayList<String> letters2 = new
ArrayList<>();
      ArrayList<Double> counts2 = new
ArrayList<>();
      for (Map.Entry<Character, Double> e:
countMap2.entrySet()) {
         letters2.add(String.valueOf(e.getKey()));
```

```
            counts2.add(e.getValue());
        }

        HashMap<String, Double> countMap3 =
charParser.createBigrams(wholeDoc);
        ArrayList<String> letters3 = new
ArrayList<>();
        ArrayList<Double> counts3 = new
ArrayList<>();
        for (Map.Entry<String, Double> e:
countMap3.entrySet()) {
            letters3.add(String.valueOf(e.getKey()));
            counts3.add(e.getValue());
        }
//        outputToEXCEL(countMap3);

        LetterChart exampleChart = new
LetterChart();
        CategoryChart chart =
exampleChart.getChart(letters, counts, letters2,
counts2);
        new
SwingWrapper<CategoryChart>(chart).displayC
hart();
    }

    public CategoryChart
getChart(ArrayList<String> letters,
ArrayList<Double> counts, ArrayList<String>
letters2, ArrayList<Double> counts2) {

        // Create Chart
        CategoryChart chart = new
CategoryChartBuilder().width(800).height(600).
title("Score
Histogram").xAxisTitle("Score").yAxisTitle("N
umber").build();

        // Customize Chart
//
chart.getStyler().setLegendPosition(Styler.Lege
ndPosition.InsideNW);
//        chart.getStyler().setHasAnnotations(true);

chart.getStyler().setLegendPosition(Styler.Lege
ndPosition.InsideNW);

chart.getStyler().setAvailableSpaceFill(.96);
        chart.getStyler().setOverlapped(true);

        // Series
//        chart.addSeries("test 1",
Arrays.asList(new String[]{"A", "B", "c", "D",
"F"}), Arrays.asList(new Double[]{4, 5, 9, 6,
5}));
//        chart.addSeries("test 1", letters, counts);
        chart.addSeries("test 1", letters, counts);
//        chart.addSeries("test 2", letters2,
counts2);


        return chart;
    }

    private static void
outputToEXCEL(HashMap<String, Double>
input) throws IOException {
        Workbook workbook = new
XSSFWorkbook(); // new HSSFWorkbook() for
generating `.xls` file

        /* CreationHelper helps us create instances
of various things like DataFormat,
           Hyperlink, RichTextString etc, in a
format (HSSF, XSSF) independent way */
        CreationHelper createHelper =
workbook.getCreationHelper();

        // Create a Sheet
        Sheet sheet =
workbook.createSheet("Grams");

        // Create a Font for styling header cells
        Font headerFont = workbook.createFont();
        headerFont.setBold(true);
        headerFont.setFontHeightInPoints((short)
14);

headerFont.setColor(IndexedColors.RED.getInd
ex());

        // Create a CellStyle with the font
        CellStyle headerCellStyle =
workbook.createCellStyle();
        headerCellStyle.setFont(headerFont);
```

```
        // Create a Row
        Row headerRow = sheet.createRow(0);

        String[] columns = new String[]{"A",
"B"};

        // Create cells
        for(int i = 0; i < columns.length; i++) {
            Cell cell = headerRow.createCell(i);
            cell.setCellValue(columns[i]);
            cell.setCellStyle(headerCellStyle);
        }

        // Create Other rows and cells with
employees data
        int rowNum = 1;
        for(Map.Entry<String, Double> inputs:
input.entrySet()) {
            Row row =
sheet.createRow(rowNum++);

            row.createCell(0)
                    .setCellValue(inputs.getKey());

            row.createCell(1)
                    .setCellValue(inputs.getValue());
        }

        // Resize all columns to fit the content size
        for(int i = 0; i < columns.length; i++) {
            sheet.autoSizeColumn(i);
        }

        // Write the output to a file
        FileOutputStream fileOut = new
FileOutputStream("C:\\Users\\DMYTROShCH
ERBAKOV\\Documents\\Projects\\study\\mdaut
or\\src\\main\\resources\\output\\output.xlsx");
            workbook.write(fileOut);
            fileOut.close();

            // Closing the workbook
            workbook.close();
        }
    }


//    HashMap<String, Double> countMap =
charParser.charParse(wholeDoc);
//
//    double[] xData = new double[] { 0.0, 1.0,
2.0 };
//    double[] yData = new double[] { 2.0, 1.0,
0.0 };
//
//    // Create Chart
//    XYChart chart =
QuickChart.getChart("Sample Chart", "X", "Y",
"y(x)", xData, yData);
//    CategoryChart categoryChart =
QuickChart.getChart("Chart", "Letters",
"Count", "Count from Letters",
countMap.keySet(), countMap.values());
//
//// Show it
//        new SwingWrapper(chart).displayChart();
//
//    int i=0;
```

## Опис модулю статистичного аналізу CharParser.java

```
package com.dip.mdautor;

import org.apache.commons.lang3.StringUtils;
import org.springframework.stereotype.Component;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.regex.Pattern;

@Component
public class CharParser {
    private Integer charsCount;

    public HashMap<Character, Double>
charParseAssimilated(String input){
        char[] chars = input.toLowerCase().toCharArray();
        chars = assimilate(chars);
        HashMap<Character, Double> countMap = new
HashMap<Character, Double>();
        for (char aChar : chars) {
            if (countMap.containsKey(aChar)) {
                countMap.put(aChar, countMap.get(aChar) +
1);
            } else {
```

```
            String tempChar1 = String.valueOf(aChar);
//
System.out.println(StringUtils.equals(tempChar1, "([A-
Z]|[a-z])"));
//              System.out.println(tempChar1.matches("([A-
Z]|[a-z])"));
              if (tempChar1.matches("([А-Я]|[а-я])")) {
                 countMap.put(aChar, 1.0);
              }
           }
        }

        //print all satisfiying max occurrence
        for (Map.Entry<Character, Double> e:
countMap.entrySet()) {
//           if (e.getValue() == max) {
           System.out.println("The number of letter " +
e.getKey() + "  is " + e.getValue());
//           }
        }
        return countMap;
    }

    public HashMap<Character, Double> charParse(String
input){
        char[] chars = input.toLowerCase().toCharArray();
        charsCount = 0;
//       chars = assimilate(chars);
        HashMap<Character, Double> countMap = new
HashMap<Character, Double>();
        for (char aChar : chars) {
           if (countMap.containsKey(aChar)) {
              countMap.put(aChar, countMap.get(aChar) +
1);
              charsCount++;
           } else {
              String tempChar1 = String.valueOf(aChar);
//
System.out.println(StringUtils.equals(tempChar1, "([A-
Z]|[a-z])"));
//              System.out.println(tempChar1.matches("([A-
Z]|[a-z])"));
              if (tempChar1.matches("(?=[А-Я]|[а-
я]|[Іі]|[Єє]|[Її])([^ЫыЪъЭэ])")) {
                 countMap.put(aChar, 1.0);
                 charsCount++;
              }
           }
        }

        //print all satisfiying max occurrence
        for (Map.Entry<Character, Double> e:
countMap.entrySet()) {
//           if (e.getValue() == max) {
              System.out.println("The number of letter " +
e.getKey() + "  is " + e.getValue());
//           }
        }

        for (Map.Entry<Character, Double> e:
countMap.entrySet()) {
           countMap.replace(e.getKey(),
e.getValue()/charsCount);
        }
        return countMap;
    }

    private char[] assimilate(char[] input){
        char[] result = input;
        for(int i=0; i<result.length; i++){
           if(result[i] == 'и') {
              result[i] = 'ы';
           }
           if(result[i] == 'ї') {
              result[i] = 'й';
           }
           if(result[i] == 'і') {
              result[i] = 'и';
           }
           if((result[i] == 'е')) {
              result[i] = 'э';
           }
           if(result[i] == 'є') {
              result[i] = 'е';
           }
        }
        return result;
    }

    public HashMap<String, Double> createBigrams(String
input){
        char[] chars = input.toLowerCase().toCharArray();
        charsCount = 0;
        HashMap<String, Double> countMap = new
HashMap<String, Double>();
        List<String> result = new ArrayList<>();
        for(int i=0; i<chars.length-1; i++){
           String tempChar1 = String.valueOf(chars[i]);
           String tempChar2 = String.valueOf(chars[i+1]);
           if((tempChar1.matches("(?=[А-Я]|[а-
я]|[Іі]|[Єє]|[Її])([^ЫыЪъЭэ])"))&&(tempChar2.matches("
(?=[А-Я]|[а-я]|[Іі]|[Єє]|[Її])([^ЫыЪъЭэ])"))){
//              result.add(tempChar1+tempChar2);
              String aChar = tempChar1+tempChar2;
              if (countMap.containsKey(aChar)) {
                 countMap.put(aChar, countMap.get(aChar) +
1);
                 charsCount++;
              } else {
                 if (tempChar1.matches("(?=[А-Я]|[а-
я]|[Іі]|[Єє]|[Її])([^ЫыЪъЭэ])")) {
                    countMap.put(aChar, 1.0);
                    charsCount++;
                 }
```

```
            }
        }
    }
    for (Map.Entry<String, Double> e:
countMap.entrySet()) {
        countMap.replace(e.getKey(),
e.getValue()/charsCount);
    }
    return countMap;
  }
}
```

## Опис модулю визначення автору FileAnalisys.java

```
package com.dip.mdautor;

import java.io.*;
import java.util.*;

import org.apache.poi.ss.formula.functions.Column;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.Row;
import org.apache.poi.xssf.usermodel.XSSFSheet;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;

public class FileAnalisys {
  static CharParser charParser = new CharParser();
  private static List<String>
tagsFinderMainTestInputArray = new ArrayList<>();
  private static String wholeDoc = "";
  private static List<String> autors = new ArrayList<>();
  private static List<List<String>> letters = new
ArrayList<>();

  public static void main(String[] args) throws
IOException {
    Boolean triggerAutor = false;
    Boolean triggerLetters = false;
    BufferedReader reader = new BufferedReader(new
InputStreamReader(
        new
FileInputStream("C:\\Users\\DMYTROShCHERBAKOV\
\Documents\\Projects\\study\\mdautor\\src\\main\\resource
s\\NechuiLevickiy\\Khmary.txt"), "UTF8"));
    String line = reader.readLine();
    while (line != null) {
      tagsFinderMainTestInputArray.add(line);
      wholeDoc = wholeDoc + line;
      line = reader.readLine();
    }
    reader.close();
    HashMap<Character, Double> countMap =
charParser.charParse(wholeDoc);

    File excelFile = new
File("C:\\Users\\DMYTROShCHERBAKOV\\Documents
\\Projects\\study\\mdautor\\src\\main\\resources\\data\\data
1.xlsx");
    FileInputStream fis = new
FileInputStream(excelFile);

    // we create an XSSF Workbook object for our
XLSX Excel File
    XSSFWorkbook workbook = new
XSSFWorkbook(fis);
    // we get first sheet
    XSSFSheet sheet = workbook.getSheetAt(0);

    // we iterate on rows
    Iterator<Row> rowIt = sheet.iterator();
    HashMap<Character, List<Double>> dataBase =
new HashMap<>();

    while (rowIt.hasNext()) {
      Row row = rowIt.next();

      // iterate on cells for the current row
      Iterator<Cell> cellIterator = row.cellIterator();
      List<String> tempLetters = new ArrayList<>();
      List<Double> tempParams = new ArrayList<>();
      Character temp = null;

      while (cellIterator.hasNext()) {
        Cell cell = cellIterator.next();
        if (triggerLetters == true) {
          tempLetters.add(cell.toString());

tempParams.add(Double.valueOf(cell.toString()));
        } else if (triggerAutor == true) {
          autors.add(cell.toString());
        } else if (cell.toString().equals("Автор")) {
          triggerAutor = true;
        } else {
          triggerLetters = true;
          tempLetters.add(cell.toString());
          temp = cell.toString().charAt(0);
        }
//          System.out.print(cell.toString() + ";");

      }
      if (tempLetters.size() != 0) {
        letters.add(tempLetters);
        dataBase.put(temp, tempParams);
      }
      triggerAutor = false;
      triggerLetters = false;

    }

    for (Map.Entry<Character, Double> e :
countMap.entrySet()) {
      List<Double> tempParams =
```

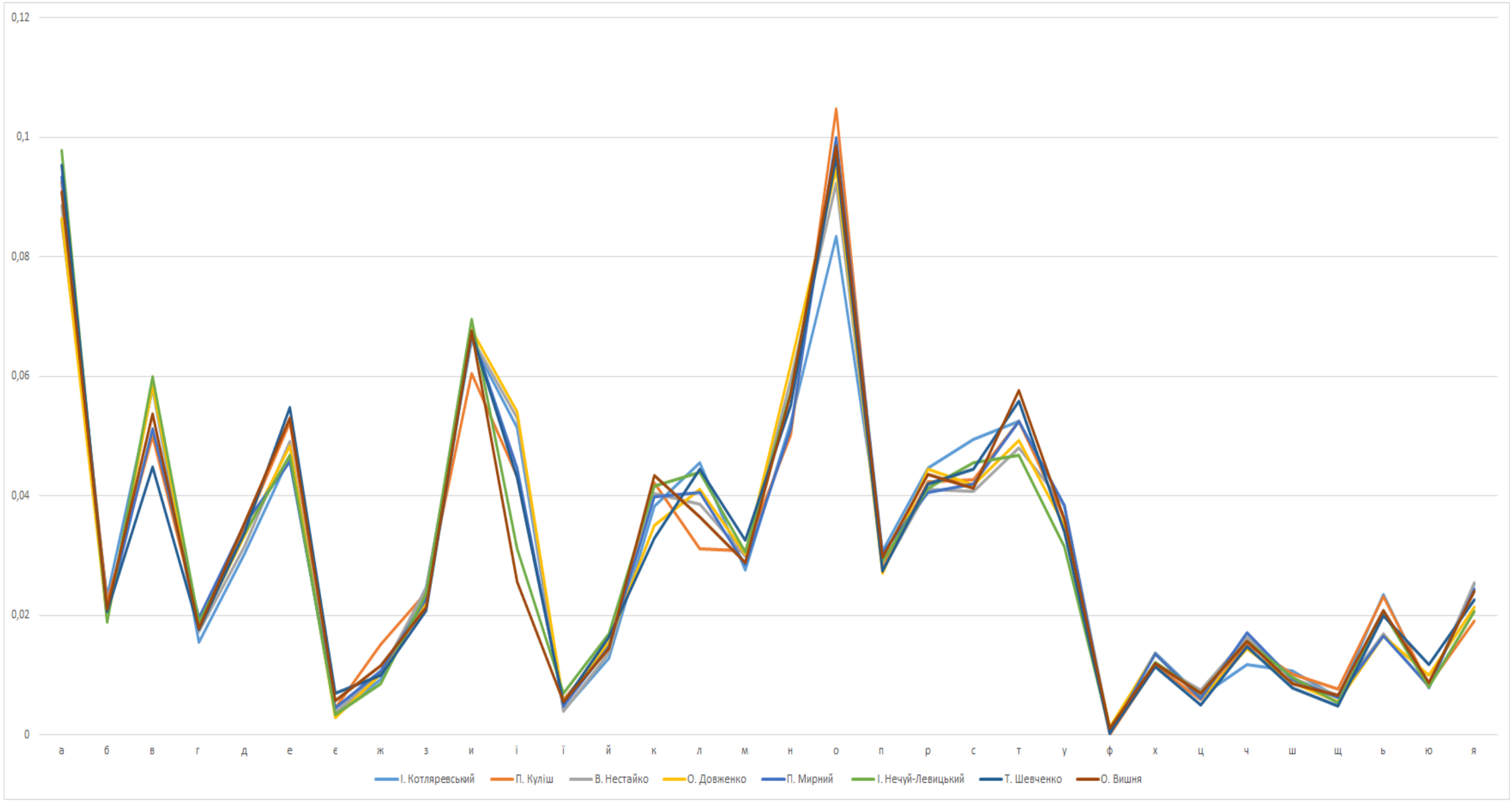
```
dataBase.get(e.getKey());
            for (int i = 0; i < tempParams.size(); i++) {
                Double tempCount = 0.0;
                tempCount = Math.abs(tempParams.get(i) -
e.getValue());
                tempParams.set(i, tempCount);
            }
            System.out.println(tempParams);
            dataBase.put(e.getKey(), tempParams);
        }
        List<Double> summarize = new ArrayList<>();
        for (Map.Entry<Character, List<Double>> e :
dataBase.entrySet()) {
            List<Double> tempParams = e.getValue();
            if (summarize.size() == 0){
                for(int i=0; i<8; i++){
                    summarize.add(tempParams.get(i));
                }
            } else {
                for(int i=0; i<8; i++){
                    if(summarize.get(i) == null){
                        summarize.set(i, tempParams.get(i));
                    } else {
                        Double tempCount = summarize.get(i);
                        summarize.set(i,
tempCount+tempParams.get(i));
                    }
                }
            }
        }
        for(int i=0; i<autors.size(); i++){
            System.out.println(autors.get(i) + ": " +
summarize.get(i));
        }


        workbook.close();
        fis.close();
    }

}
```

# ДОДАТОК Б ГРАФІЧНИЙ МАТЕРІАЛ

# ДОДАТОК Б1 ГРАФІК ЧАСТОТНОГО РОЗПОДІЛУ ГРАМ АВТОРІВ УКРАЇНСЬКОЇ ЛІТЕРАТУРИ



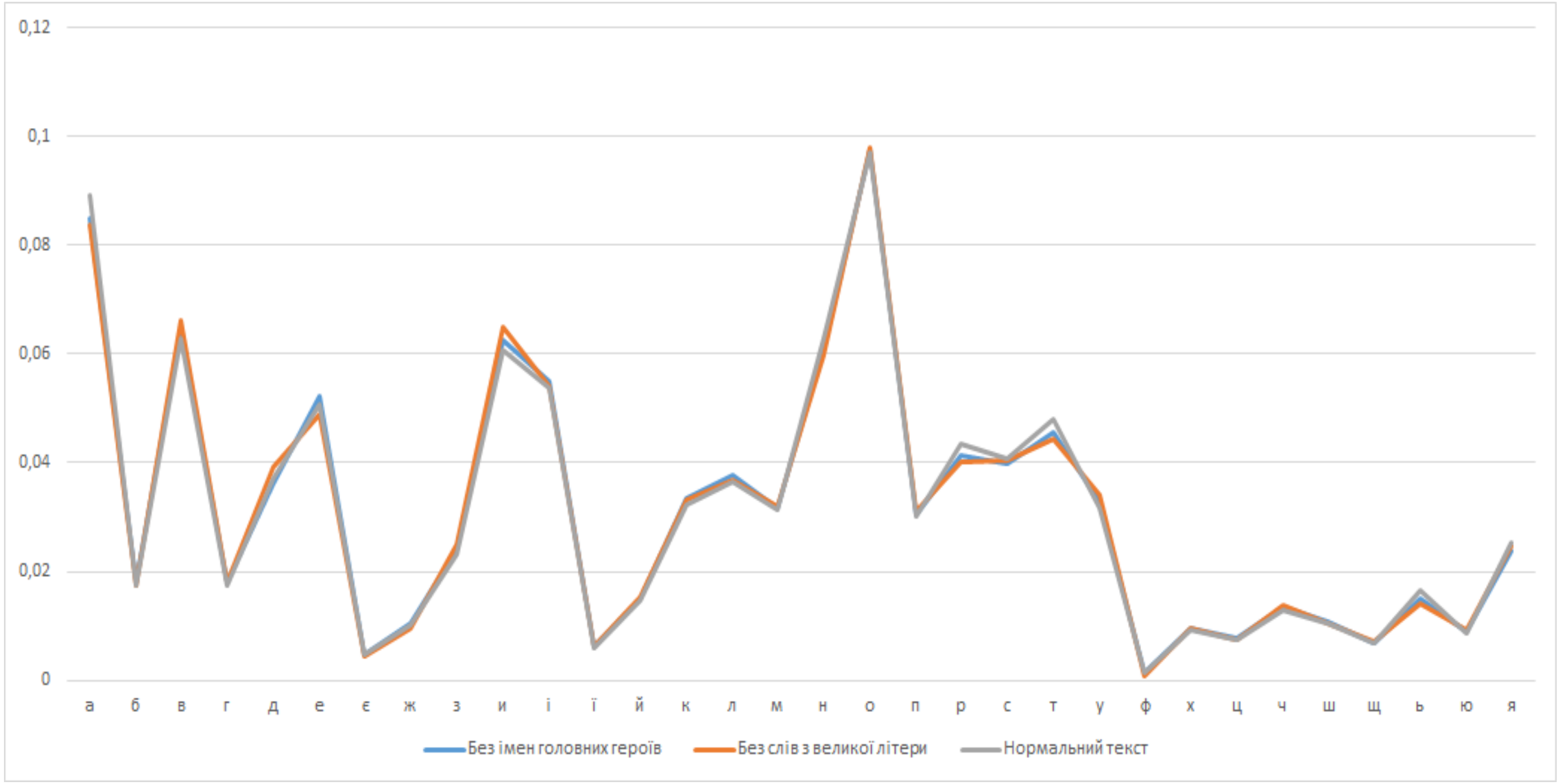

Демонстраційний плакат до магістерської дисертації

Графік частотного розподілу грам авторів української літератури

Виконав студент гр. ІП-82мп Щербаков Д.С.

Керівник Фіногенов О.Д.

# ДОДАТОК Б2 АНАЛІЗ ВПЛИВУ ІМЕН ГОЛОВНИХ ГЕРОЇВ НА РОЗПОДІЛ ГРАМ




Демонстраційний плакат до магістерської дисертації

Аналіз впливу імен головних героїв на розподіл грам

Виконав студент гр. ІП-82мп Щербаков Д.С.

Керівник Фіногенов О.Д.